OPTIMAL COLLISION-FREE PATH PLANNING

by

COMMURI SESHADRI

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

FEBRUARY, 1989

OPTIMAL COLLISION-FREE PATH PLANNING

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by

COMMURI SESHADRI

to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

FEBRUARY, 1989

CERTIFICATE

This is to certify that this thesis titled "OPTIMAL COLLISION - FREE PATH PLANNING" has been carried out by Commuri Seshadri under my supervision and the same has not been submitted elsewhere for a degree.

AGhosh
(Dr. Arindam Ghosh)

Asst. Professor.
Department of Electrical Engineering
Indian Institute of Technology.
KANPUR

ACKNOWLEDGEMENTS.

It is my pleasant duty to express my sincere gratitude to Dr.Arindam Ghosh for his encouragement and unfailing guidance during the course of this work. His enthusiasm and cooperation have been the mainstay of this work. I am also indebted to my parents who have always been an unending source of inspiration and motivation.

I take this opportunity to thank my friends V.V.Ram Prasad, P.Suneel, S.Majee and Satish Kaveti for helping me in many ways during this work.

COMMURI SESHADRI.

ABSTRACT.

Optimal collision - free path planning for robot manipulators has been attempted utilizing the method of Local Variations (MLV). Methods for checking collision between robot arm and static, as well as dynamic, obstacles have been developed. Collision is detected by checking for geometric intersection of two objects in the workspace of the robot. This scheme has been incorporated in the MLV algorithm and used to solve the minimum - energy path planning problem for a robot in the presence of static obstacles. This approach has also been applied to solve the minimum - energy and minimum - time path planning problems for two robots operating simultaneously in the same workspace.

A scheme for determining the changes in energy with variation in traversal time has been developed. This scheme is utilized to determine the minimum - time - energy geometric path for a single robot. All these methods were implemented on the DEC - 1090 mainframe computer and the results verified through computer simulation of the first three joints of PUMA - 560 manipulator.

CONTENTS.

CHAPTER 1.

INTRODUCTION.

The requirement of consistent quality together with high productivity has led to rapid increase in the use of robotic manipulators in industry. An industrial robot is a multifunctional programmable manipulator which can perform flexible tasks. Robot systems are capable of performing various tasks requiring high degree of dexterity and adaptability to various environments with relatively low costs.

With increasing stress on productivity and optimum utilization of resources, task planning and control of robotic manipulators attain significance. Some of the factors to be considered during task planning are the energy consumption and the time of operation. Further, there may be a number of static, as well as dynamic obstacles in the workspace of the robot. The control of a robotic manipulator should be such that it drives the robot arm along a path free from collisions while consuming minimum amount of energy or traveling in a minimum time.

1.1. Path Planning And Control :

Robots must be able to work in large crowded spaces, handle a variety of work pieces and perform flexible tasks. The dynamic behavior of robotic manipulators is highly complex, since the dynamics of multi-input, multi-output spatial linkages are highly coupled and nonlinear.

The kinematic and dynamic complexities create an unique control problem that cannot be adequately handled by standard linear control techniques. This makes effective control system design a critical and exciting issue in robotics.

Further, robots are required to interact much more heavily with peripheral devices than traditional numerically controlled machine tools. These machine tools are essentially self contained systems that handle work pieces in well defined locations. By contrast, the environment in which robots are used is often poorly structured and effective means must be developed to identify the locations of workpieces as well as to communicate with peripheral devices and other machines in a coordinated fashion.

In general, it is extremely difficult to obtain an exact closed-form solution to the optimal control problem of robots mainly due to the following reasons:

(i) the non-linearity and coupling in the manipulator dynamics and

(ii) the complexity involved in collision avoidance computations.

Due to this, the optimal control problem is usually divided into two subproblems, i.e. trajectory planning and trajectory tracking. This division can be easily explained by Fig 1.1.[1]

Thus, the optimal control problem is to determine the controls which will drive the manipulator

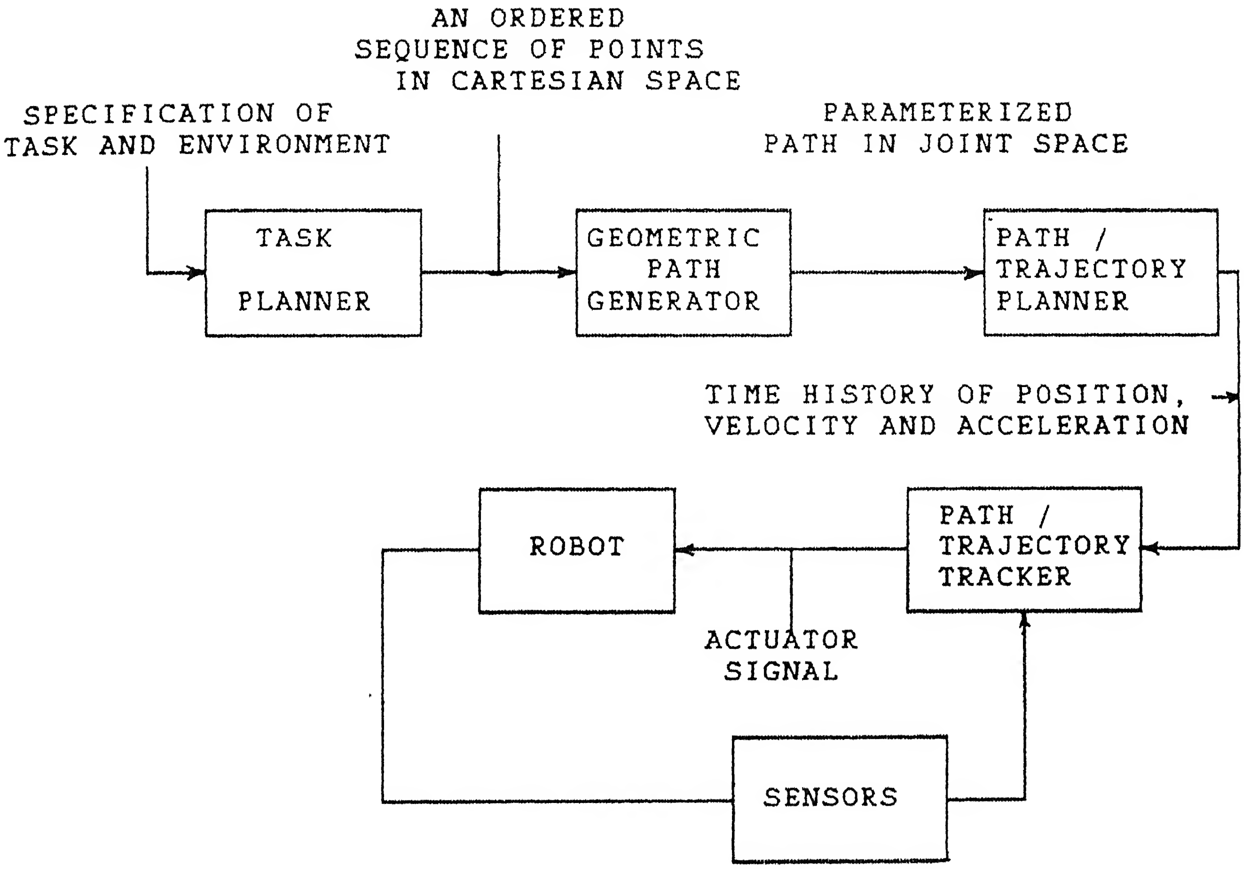


FIG.1.1 FUNCTIONAL BLOCK DIAGRAM

OF ROBOT POSITION CONTROL

from a given initial configuration to a given final configuration under the actuator and positional constraints, while minimizing a preselected cost function.

1.2. Earlier Approaches To Path Planning :

A number of model based manipulator programming problems have been addressed independent of any manipulator system, especially the problem of collision detection and collision avoidance among obstacles. The earliest algorithms first formulated the problem of collision avoidance in terms of an obstacle transformation which allows the moving object to be treated as a point. This is termed as the configuration space approach. [2 - 5].. Conceptually, the configuration space approach may be viewed as shrinking the object to a point while at the same time expanding the obstacles to the shape of the moving object. This approach works well when the moving object is not allowed to rotate. If it rotates, then the grown obstacle must be embedded in a higher dimensional space - one dimension for each degree of rotational freedom. Furthermore,the grown obstacles will have nonplanar surfaces even when the original surface was polygonal or even polyhedral. This problem was partly overcome by slice projection method [6]. Here the rotation range is split into a fixed number of slices and within each slice the object is bounded by a polyhedra.

In the Visibility graph method [2,4],

the robot is represented as a point and the obstacles are suitably grown. From the initial point on the path, the final point is viewed. If this point is visible, it indicates the presence of collision-free path which is a straight line joining the initial and final points of the desired robot motion. If the goal is not visible, the robot is moved to the nearest vertex of the obstacle, which is lying on the line joining the initial and final points of the desired path. The robot moves along one of the edges of the obstacle till it reaches another vertex or till the goal is visible. This method, usually used in navigation, is used to compile the visibility graph. Collision-free trajectories can be developed by searching the visibility graph.

Both the configuration space and visibility graph approaches deal with the problem of find space and find path, i.e. problems that involve finding where to place or how to move a solid in the presence of obstacles.

Boyce [7] presented a structure for representing three dimensional objects and an algorithm for detecting intersections and collisions among these objects. Three basic object types are represented, viz. surfaces, solids and containers. To simplify interference checking computations, computer representations are limited to polyhedra. Static interference detection is done by checking for surface intersections, while dynamic collision

is checked by finding out intersection between two faces or between a face and an edge. Brooks [8,9] characterized 'Free space' as an union of generalized cones. A collision-free path for convex polygonal bodies through space littered with obstacle polygons can be got by maximizing the distance of closest approach to an obstacle. The method is based on characterizing the volume swept by a body as it is translated and rotated as a generalized cone while determining the conditions under which it is a subset of another cone.

Chien [10] utilized the concept of state space and rotational mapping graphs to determine collision- free paths. The relationship between the position and the corresponding collision-free orientation of a moving object among obstacles is represented as a set of state space. This set is called 'Rotational Mapping Graph'. The problem of finding collision-free path for an object translating and rotating among obstacles is solved considering the connectivity of the ' Rotational Mapping Graph '.

Hasegawa and Terasaki [11] solved the problem of collision avoidance by characterizing a ' Fully-Free Space' where the robot can assume any configuration. Hence, the path has to be carefully chosen only in the vicinity of the obstacles, as any configuration is possible in the fully-free space.

Most of the collision avoidance schemes mentioned emphasize a single robot within a fixed environment of stationary obstacles and achieve varied

degrees of success. Of the few researchers who addressed the problem of path planning for multi-robots, Lee and Lee [12] proposed a method for determining potential collisions between the wrists of two robots while moving on straight line paths. Collisions between the wrists of the two robots is avoided by proper time scheduling.

Nagata et.al [13] proposed a robot plan generation system which treats continuous state changes in time for multiple robots. The plan generation system consists of two subsystems - a fundamental planning subsystem for multiple robots and a system for detecting and avoiding mutual collisions of robots. In the strategy for avoiding collisions, the arms are modeled as cylinders. The collisions are avoided by the following three methods :

(i) When both the robots are to access a close point,only one robot acts at a time while the other robot remains idle.

(ii) Collisions in movement are avoided by moving the arms in horizontal planes separated by a suitable distance.

(iii) When (i) and (ii) are not feasible, then an exact check for detecting collisions between the two arms is performed.

The first subsystem obtains the fundamental sequence of actions while the second subsystem detects and avoids mutual collisions of robots. The final planning is performed by modifying the former subsystem through the

results of the later subsystem.

Most of the methods discussed above work satisfactorily where collision detection and avoidance are the primary concerns. But , when the requirement is an optimum collision-free path, then these methods are inadequate. Gilbert and Johnson [14,15] presented an algorithm which utilizes the concept of distance function to minimize the deviation from the center line path between obstacles. Suh and Shin [16] utilized a similar approach to determine a channel in the workspace of the robot, which contains at least one feasible path among the obstacles. A variational dynamic programming approach with minimum distance criterion is used to find the feasible path. However, very few researchers have tackled the problem of finding collision-free minimum-time or minimum-energy paths for robots.

Among the optimum path planning problems, the minimum-time path planning problem (MTPP problem) has received wide attention. The reason for this is that for economical use of robots, they should be operated to maximize the number of jobs performed in a given time. The first reported work dealing with MTTP problem is by Kahn and Roth [17]. Their formulation resulted in a highly nonlinear optimal control problem , which is difficult to solve. Luh and Walker [18] and Luh and Lin [19] attempted to obtain minimum time along a given path by applying a combination of linear and nonlinear programming. These works assumed that the desired path is specified in terms of the two end points

and a set of intermediate points, called corner points. The robot's end effector is moved in straight line segments between the corner points with uniform velocity and in circular arcs around the corner points with uniform acceleration. They neglected the robot dynamics and put constraints on the maximum linear and angular velocities and accelerations. However, these extremal bounds depend on robot configuration, load and several other factors. It has been demonstrated that to accommodate the constraints on the accelerations, the velocities rarely reach their maximum specified values resulting in under utilization of the robot.[20]. Moreover, as the trajectory is generated in the Cartesian space, it is difficult to check the violation of actuator limits.

One of the major break throughs in the time optimal control problem was reported by Hollerbach [21]. In this work, the time scaling property of manipulator dynamics which allowed modification of movement speed without recalculation of complete dynamics was proposed. Two elegant and somewhat similar solutions to the MTTP problem have been reported by Bobrow [22,23] and Shin and McKay [24]. The path to be followed by the robot is considered as a parameterized curve, and the optimal solutions are given by switching curves in phase plane. In [22], the actuator torque limits are functions only of robot positions and velocities. Shin and McKay on the other hand, considered the torque bounds as quadratic functions of the joint velocities.

A minimum-time trajectory is not always cost efficient. Moreover, the two methods mentioned above cannot take into consideration constraints on jerks, which may be required to prevent excessive wear on the mechanism. Shin and McKay [25] presented a method to tackle these limitations. This method, which solves a path-constrained-time-energy problem uses dynamic programming and is computationally expensive.

The desired paths along which a robot must travel is usually specified in the Cartesian coordinate space and then converted, point by point, into the joint space of the robot. The minimum time of traversal depends on the geometric path selected. Shin and McKay [1] have presented a method for selecting near-minimum-time geometric path.

All the methods presented assume that a collision-free path is already known, and the only requirement is to minimize the traversal time along this path. However, when a collision-free path is to be chosen such that a task can be performed in minimum time,then these methods are not useful. Furthermore, these methods rarely handle the minimum - time - energy problems. Patrikar [26] solved the problem of minimum-time path planning, as well as, the minimum-energy path planning in a novel fashion utilizing the 'Method of Local Variations' [27,28] and the parameterization approach of Bobrow et.al [23]. The minimum-energy, as well as, the minimum-time path planning problems were solved in a computationally efficient and attractive

manner. This method finds an optimal trajectory by perturbing a starting nominal trajectory. One of the shortcomings of this work is that it assumes that obstacles are specified by bounds in the joint space. This however, need not be the case as most of the obstacles are specified in the Cartesian coordinate space.

1.3 Objective of the thesis :

In this thesis, the method of Local Variations (MLV) has been employed to solve a variety of robot path planning problems. Methods for collision checking for static, as well as, dynamic obstacles are presented. These methods detect collision by computing the distance between the objects in the three dimensional space. The Euclidean metric is used because it conforms with the physical notion of distance and makes it invariant with respect to different choices for the origin and orientation of the coordinate systems. The collision checking method is incorporated into MLV and three types of robot path planning problems are proposed.

First, a minimum-energy path is obtained in the presence of static obstacles. Then, minimum-energy paths for two robots working simultaneously is solved. The path planning for each robot treats the presence of the other robot as a dynamic obstacle to be avoided. In the third problem, specified the collision-free paths for two robots, the minimum time of traversal for each robot is determined.

In addition to the above three problems, the

problem of selecting a near-minimum-time-energy geometric path for a robot is also proposed.

1.4 Breakdown of Chapters :

Chapter 2 deals with 'Collision-free minimum-energy path planning ' for a single robot. In this chapter, the optimal control problem has been formulated using robot dynamics. A method for checking collisions with static obstacles is developed. The problem has been solved by application of MLV. Results obtained through digital computer simulations are presented to verify the problem formulation.

Chapter 3 deals with ' Collision - free minimum - energy paths ' for two robots. A method for detecting collisions between two robot arms is presented. Two different types of problem formulations are considered. In the first case, a minimum-energy path for one robot is obtained. Given this information, a minimum-energy path for the second robot is obtained while avoiding any collisions with the first robot. In the second case, minimum-energy paths for both the robots is obtained simultaneously. Numerical examples are solved using MLV and the results obtained through digital computer simulations are presented.

Chapter 4 deals with ' Minimum - time path planning ' for two robots. The parameterized robot dynamics is used to formulate this problem. The collision checking scheme developed in Chapter 3 is utilized for solving the minimum-time path planning problem for the two robots.

In Chapter 5, the formulations of Chapter 2 as well as, Chapter 4 have been utilized to develop the formulation for obtaining near - minimum - time - energy geometric paths. Different starting trajectories are considered and some guidelines have been formulated for finding near-minimum - time - energy paths.

Chapter 6 lists the various conclusions drawn through simulation studies and identifies areas where further work needs to be done.

CHAPTER 2

COLLISION-FREE MINIMUM-ENERGY PATH FOR A SINGLE ROBOT

In an industrial environment where a robot is required to perform repetitive jobs, the problem of selecting an optimum path attains significance. Since the workspace of the robot may contian various obstacles, the path must be so chosen that the arm avoids collision with static obstacles present in the workspace.

Given the initial and final end-effector configurations and the time of travesal, from the efficiency point of view, it is required to find a path along which teh energy consumption will be minimum. At the same time, the robot must avoid collision with any obstacle. This is called an obstacle-free minimum-energy path.

Most of the path pllanning schemes deal with the problem of avoiding static obstacles in the workspace of the robot. Due to this, the problem is usually cpnverted into a geometric analysis problem for checking collisions. Ususally, most of the static obstacles are represented by regular polyhedra. Exact representation is often unnecessary and increases computation. For collision detection, the primary requisite is a simple representation of the manipulator which eases the job of collision checking. It is sufficient to assume that any collision can occur between an obstacle and the wrist or the third llink of the robot. To simplify the representation, the wrist is modeled as a sphere

and the third link is modeled as a cylinder. Collision detection is done by checking for actual intersection between the arm and the obstacle. This Chapter deals with collision-free minimum-energy-path-planning for a single robot using the method of Local Variations.

2.1 Problem Formulation :

Neglecting the actuator inertia, the torque acting on a manipulator joint is given as

$$u_i = \sum_{j=1}^{n} D_{ij}\,\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} H_{ijk}\,\dot{q}_j\,\dot{q}_k + G_j \qquad i=1,\ldots,n \qquad (2.1)$$

where 'n' is the total number of joints. The first term after the equal to sign is the inertia related term, the second one is due to centrifugal and coriolis forces acting on the torque and the last one is due to gravitational forces. Defining

$$\left.\begin{aligned} X_1 &= \dot{q}_1 \\ &\;\,| \\ X_n &= \dot{q}_n \end{aligned}\right\} \qquad (2.2\text{ a})$$

$$\left.\begin{aligned} X_{n+1} &= q_1 \\ &\;\,| \\ X_{2n} &= q_n \end{aligned}\right\} \qquad (2.2\text{ b})$$

Eqn. (2.1) is rewritten as

$$u_i = \sum_{i=1}^{n} D_{ij}\,\dot{X}_j + \sum_{i=1}^{n}\sum_{i=1}^{n} H_{ijk}\,X_j\,X_k + G_j \qquad (2.3)$$

$$i = 1,\ldots, n$$

The state vector defined by eqn.(2.2) has 'n' elements, i.e., the positions of each joint (q_i). To minimize the energy along the path of travel a cost function is chosen as

$$J_E = K_1 \int_0^{t_f} KE\ dt + K_2 \int_0^{t_f} PE\ dt \tag{2.4}$$

where KE and PE are the kinetic and potential energies respectively. K_1 and K_2 are two weighting factors. It is assumed that the time of traversal t_f is fixed and the initial and final configurations are known.

The MLV is then employed to find the optimal trajectory and its associated control. To do that first the time interval $[0,t_f]$ is divided into N equally spaced subintervals such that

$t_f = N\delta T$

An initial nominal trajectory $X^1(k),(k=1,..,N)$ is considered. The incremental cost between 2 successive time instants is calculated as

$$F_l = K_1 \int_{l-1}^{l} KE\ dt + K_2 \int_{l-1}^{l} PE\ dt \tag{2.5}$$

If δT is sufficiently small, then eqn.(2.5) can be approximated as

$$F_l = K_1\ \delta T\ KE\,\Big|_{k=l} + K_2\ \delta T\ PE\,\Big|_{k=l} \tag{2.6}$$

The values of the cost functional in each of the subintervals is computed and stored.

The optimal cost is obtained by successive perturbation of each of the components of the state vector. The variation step size is fixed for each state component. A variation in the state at K^{th} instant results in changes in position in that instant and changes in velocity and acceleration in the subsequent time instants. After each variation, tests are performed to check whether

(i) the state is admissible

(ii) the resulting position does not result in collision with any of the obstacles

(iii) the control inputs required to translate the states from X(k-1) to X(k+1) via X(k) are admissible

(iv) the subinterval costs for the intervals k-1 to k+1 is less than the cost before variation.

If the above tests give positive results, then the original state at instant k is replaced by the perturbed value of the state.

The local variation operation is applied successively for all components of the state vector at all instants of time to complete one iteration. At the end of one iteration, a new nominal trajectory is got, which is used as the starting trajectory for the next iteration. If at the end of any iteration there is no reduction in cost, then the perturbation step size is reduced by 50% and the process continued. This process which ensures a reduction in overall

cost is performed till a satisfactory convergence is reached. The resulting $X^*(k),(k=1,...,N+1)$ and $u^*(k),(k=1,..,N)$ are the desired optimal state and control vectors respectively.

2.2 Collision Avoidance Scheme :

Consider a 6 DOF robot with all revolute joints. Since most manipulators are of wrist partitioned type, the robot can be decomposed into 2 parts - an arm and a wrist, each containing 3 DOF. [12] Again as most of the obstacles are specified in the Cartesian coordinates, a simple model of the robot is necessary to simplify geometric constraint checking. For this, the wrist is modeled as a sphere, and the third link modeled as a cylinder. This arrangement is shown in Fig 2.1. It is assumed that the robot is going to be placed in such a way that there can be no collision between the second link and the obstacle.

The calculation of exact obstacle interference is computationally intensive. Hence, some simplification is made in obstacle representation. As most of the obstacles found in any workspace are of planar, cylindrical etc., the obstacles are represented by regular polyhedra. To simplify the problem still further, any static obstacle present is assumed to be in the form of a wall. The robot has to travel from one side of the wall to the other side.

Referring to Fig 1, once the joint angles of the robot arm are known at any given time, the coordinates of points A and B can be calculated through forward kinematic relations. The obstacle is shown in Fig 2.2. Suppose the

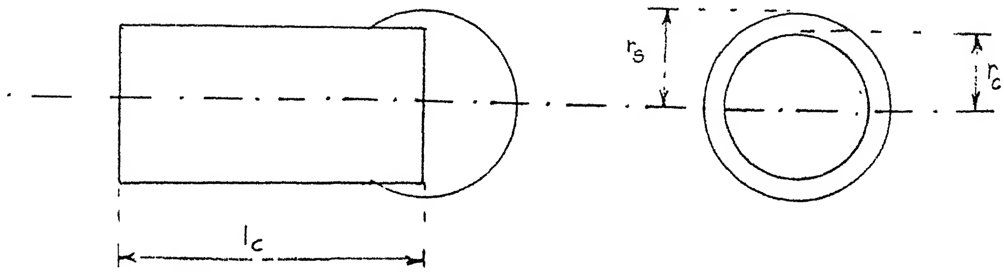


Fig. 2.1 : Simplified model of the robot.

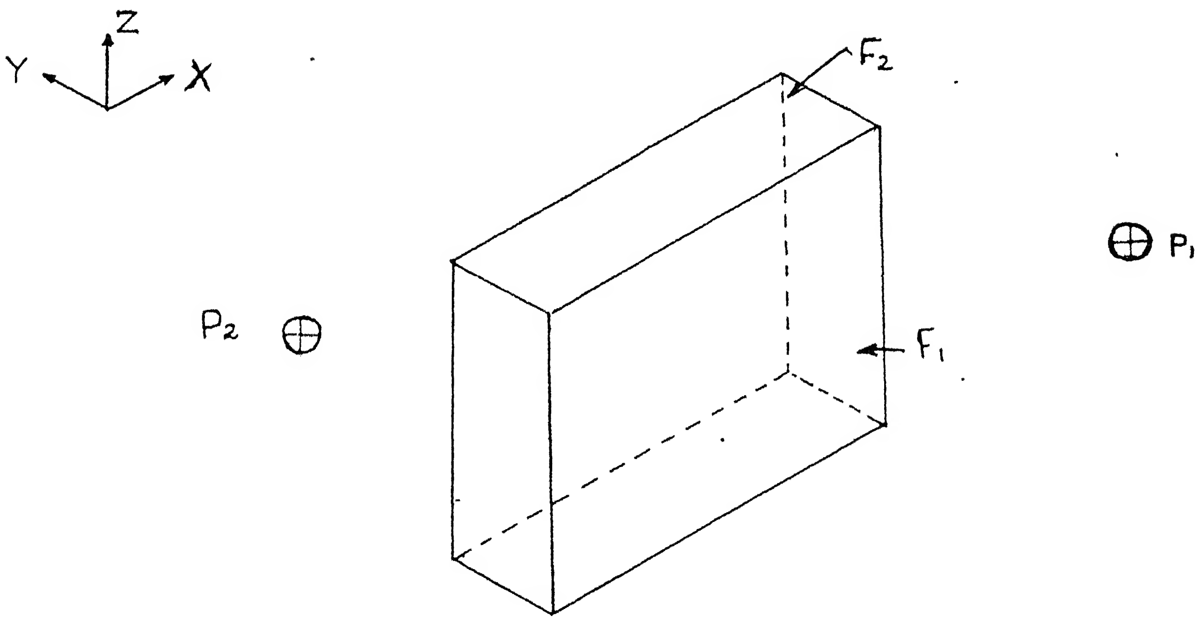


Fig. 2.2 : Layout of the obstacle.

robot has to move from a point P_1 to a point P_2 each placed on either side of the wall (Refer Fig 2.2). Then it must avoid collisions with faces F_1 and F_2. Thus after every local variation, potential collision between any of these two faces and the robot have to be checked. This is done in two stages. First a collision check with the wrist is made, and then if found collision free a collision check is made with the third link. The variation is accepted if both these collision checks are passed.

2.2.1 Collision checks for the wrist :

The collision check is easier for the wrist than for the last link. Suppose the face with which the collision check is currently made is given by PQRS, and the center of the sphere is a point A (see Fig 2.3). Then, the plane on which the face is lying is determined from the coordinates of any three of the vertices of PQRS. Let the equation of the plane be given by

$$ax + by + cz + d = 0 \qquad (2.6)$$

Next step is to find out the distance between the center of the sphere and this plane. Let (x1,y1,z1) be the coordinates of the center of the sphere. Then this distance is given by

$$\text{distance} = \left| \frac{ax1 + by1 + cz1 + d}{\sqrt{(a^2+b^2+c^2)}} \right| \qquad (2.7)$$

If this distance is less than or equal to the radius of the sphere(distance $\leq r_s$), then a potential for collision exists. This is not sufficient to prove that collision will occur

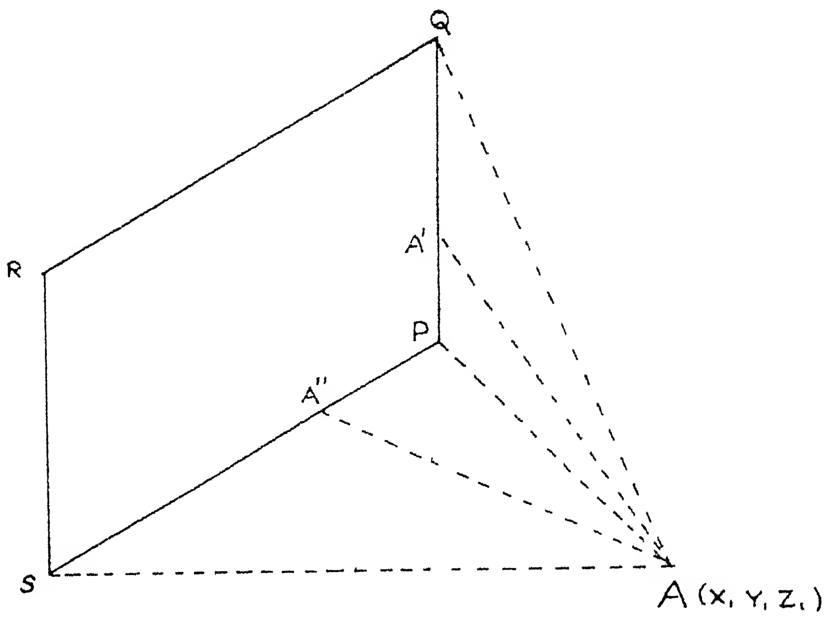


Fig. 2.3 : Collision check for the wrist.

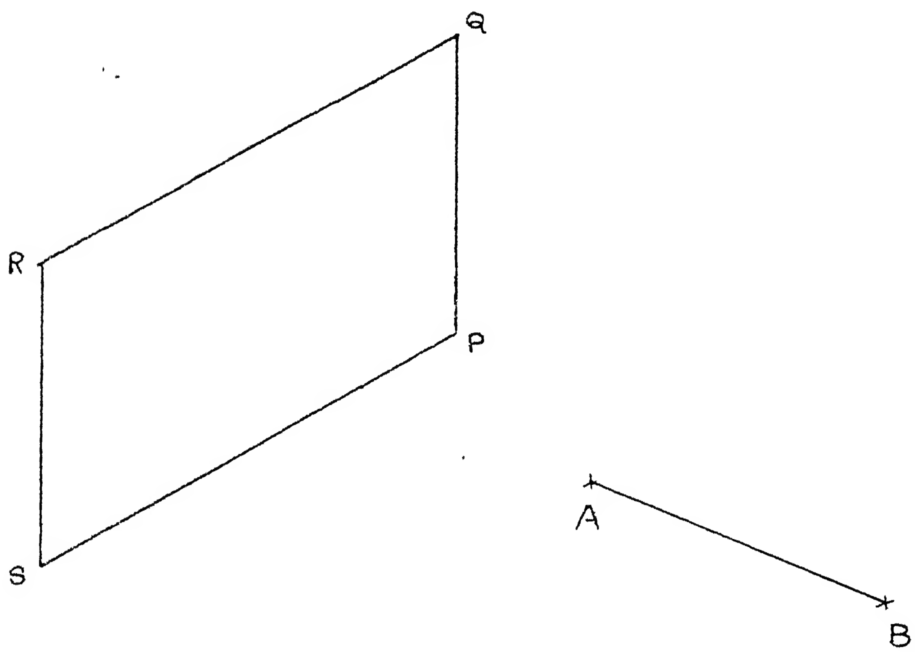


Fig. 2.4 : Collision check for the third link.

because the face is contained in a finite portion of the plane.

If a potential for collision exists, then the positive occurrence of collision must be checked. To do that, first point A is joined with any vertex of the face, say point P. Then the line PA is projected onto the straight line joining PQ, i.e., one edge of the face., The same step is repeated for the line containing edge PS. Now, if the point A is inside the face, then the projected lengths will be less than the length of the edges, and the collision is going to occur if the following conditions hold.

$$\begin{aligned} PA' + QA' &< PQ + 2r_s \\ PA'' + QA'' &< PS + 2r_s \end{aligned} \qquad (2.8)$$

where A\` and A\`\` are the projections of point A on the lines PQ and PS respectively.

2.2.2 Collision Check for the third link :

Knowing the forward kinematic relations, the coordinates of points A and B are calculated which are the end points of the line passing through the center of cylinder (see Fig 2.4). Let (x\`,y\`,z\`) be any point on this line. Then the equation of this line is

$$\frac{x'-x1}{l} = \frac{y'-y1}{m} = \frac{z'-z1}{n} = r \qquad (2.9)$$

where (x1,y1,z1) are the coordinates of point A, l,m,n are the direction cosines and r is a constant. Solving eqn (2.9)

$$x` = x1 + lr$$
$$y` = y1 + mr \qquad (2.10)$$
$$z` = z1 + nr$$

Again, the distance between the point (x`,y`,z`) and a plane containing the face of interest is given by

$$\text{distance} = \left| \frac{ax` + by` + cz` + d}{\sqrt{(a^2+b^2+c^2)}} \right| \qquad (2.11)$$

Let this distance be equal to the radius of the cylinder r_c. Then substituting equation (2.10) in equation (2.11) r can be calculated which will lead to the determination of the coordinates (x`,y`,z`). The point (x`,y`,z`) is at a distance of r_c from the plane containing the face of interest.

If the point (x`,y`,z`) lies within the points A and B then there exists potential for collision. However, to be sure, it is to be ascertained whether the projection of this point lies within the face of interest. This is done using the method described earlier.

2.3 Numerical Results :

The collision checking scheme is incorporated into the method of local variations. This algorithm is tested through digital computer simulation of PUMA 560 robot arm. The results obtained are discussed below.

Example 2.1 : The robot arm is to travel from a point (0.6,0.2,0.3) to a point (0.7,-0.2,0.2) in the Cartesian space. These two points are on either side of a static

obstacle which is a wall. The obstacle is aligned along the X axis in the Cartesian space and has a length of 0.8 m, height of 0.3 m and thickness of 0.01 m.

The robot has to travel in a optimal path such that the cost of equation (2.4) is minimized. Hence the value of K_1 is chosen as 15 and that of K_2 is chosen as 0.5. This is to ensure that the Kinetic Energy term and the Potential Energy term are of the same order. It is also assumed that any violation of torque bounds can occur only in first 3 joints. The basis for this assumption is that the wrist is rotationally invariant and hence detailed torque computation for the last three joints is not required. The torque bounds for the first 3 joints are taken as

$$|u_1| \leq 200 \text{ Nm}, \quad |u_2| \leq 400 \text{ Nm}, \quad |u_3| \leq 200 \text{ Nm} ;$$

First a nominal feasible trajectory is chosen such that its projection on both XY and YZ planes is a semicircle. The robot is assumed to have left-above arm configuration. Fig 2.5 depicts the projection of the third link and the wrist on the YZ plane for some discrete points along the nominal trajectory. The maximum clearance above the obstacle is 0.25m. along the Z-axis. The cost of this trajectory is calculated as 119.83 units.

The MLV is now applied and the projection of the third link and the wrist on YZ plane for some discrete points along the optimal trajectory is shown in Fig 2.6. The maximum clearance thus obtained is 0.1 m., which is a

reduction of 0.15 m. in height. The total cost is reduced to 82 units. The obstacle is now removed and MLV applied to find an optimal path for this case. The projection for different points is shown in Fig 2.6 with squares. It can be seen that the optimal trajectory passes directly through the area where the obstacle would have been.

Fig 2.7 shows the projections of all the three trajectories on XY plane. It can be seen from this figure that the trajectory obtained for the obstacle removed case is a line with minimum curvature, while the optimal trajectory in presence of obstacle is a curved path so as to avoid colliding with obstacle. The behavior of these trajectories are evident from Fig 2.8 and Fig 2.9 which plot the joint angles for joints 2 and 3, respectively. It can be seen that the optimal joint trajectory in the presence of obstacle is almost two straight lines drawn from two extreme points, while it is almost a straight line joining the two extreme points when the obstacle is absent.

2.4 Conclusions :

In this chapter a scheme for checking collisions of the robot are with the static obstacles in the workspace has been presented. This method had been incorporated into the MLV algorithm in order to obtain collision free minimum energy paths. The collision checking schemes presented are rather stringent. However, they reduce computational requirements. The algorithm is tested through digital computer simulations.

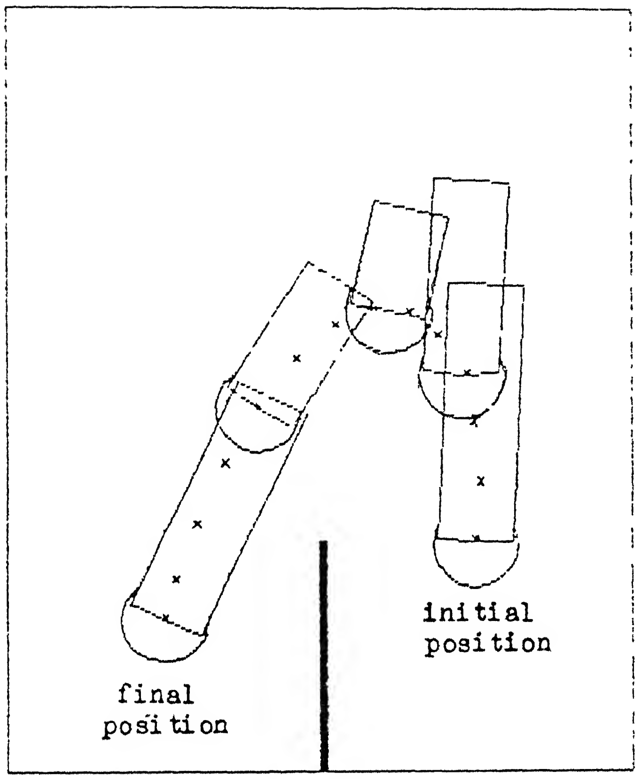


Fig. 2.5 : Nominal trajectory of the arm in Y-Z plane.

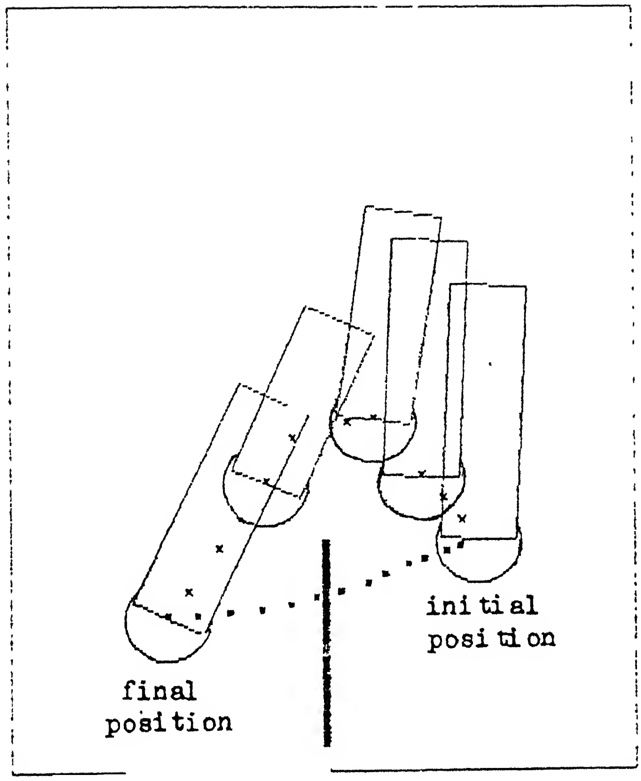
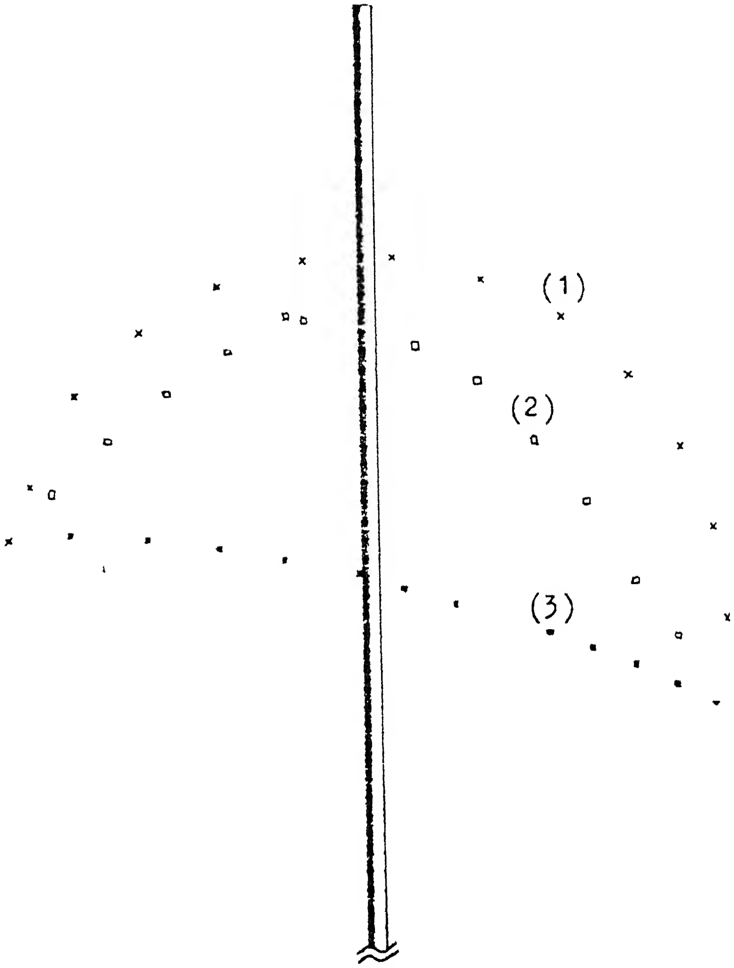


Fig. 2.6 : Optimal trajectory of the arm in Y-Z plane.



(1) Nominal trajectory
(2) Trajectory with obstacle
(3) Trajectory without obstacle

Fig. 2.7 : Projections of the trajectories on X-Y plane.

FIG.2.8. JOINT 2 POSITION PLOT.

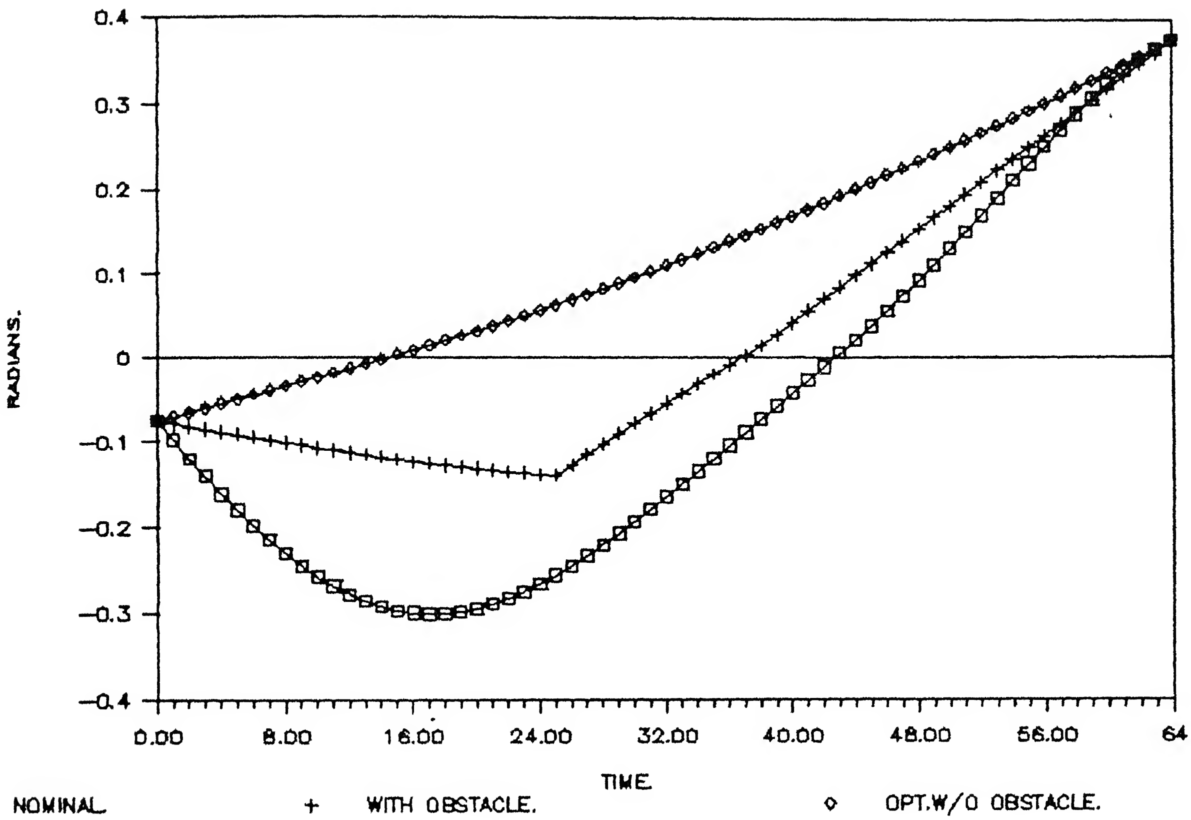
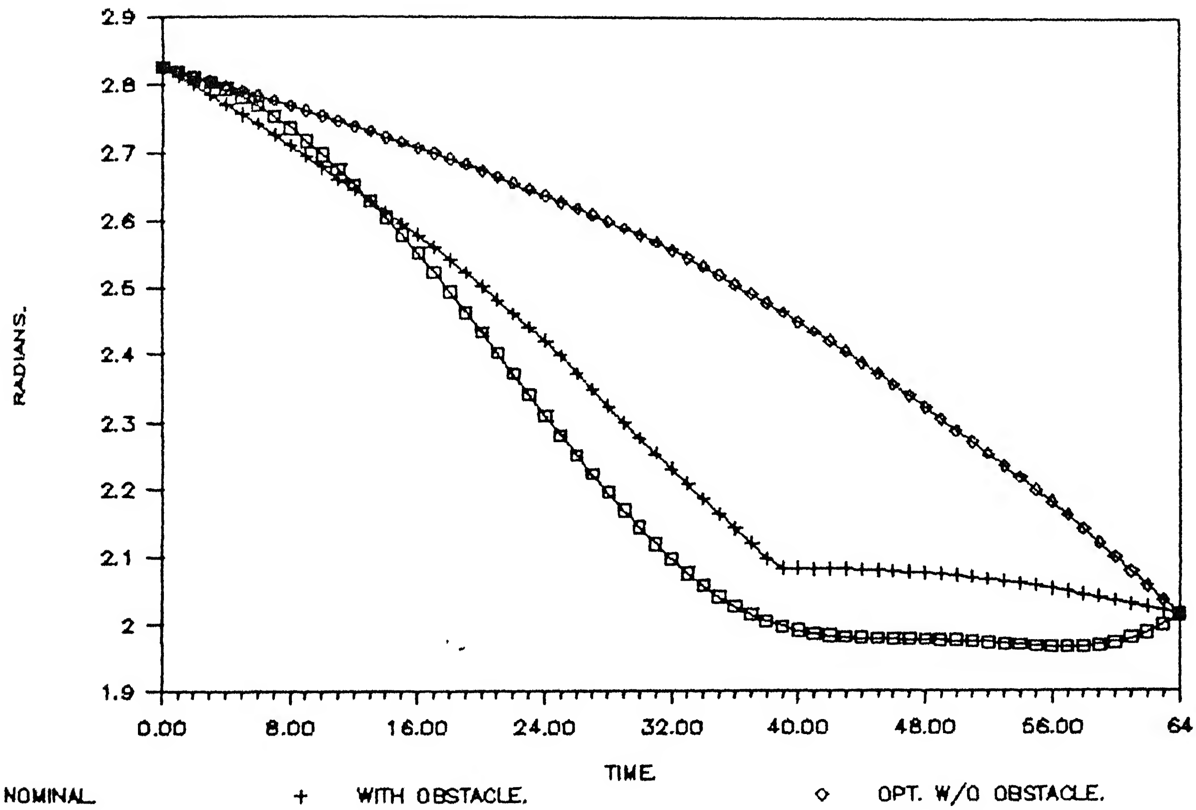


FIG 2.9. JOINT 3 POSITION PLOT.



An important point to be observed is that any variational method can converge to local optima. To overcome this problem, it may be necessary in practice to obtain several solutions with different starting trajectories before drawing any conclusion on any optimal path.

CHAPTER 3.

COLLISION-FREE MINIMUM-ENERGY PATH PLANNING FOR TWO ROBOTS.

In any industrial environment, the jobs to be performed may involve too many processing steps or simply the job may be too complicated for a single robot to handle. In such circumstances it may be necessary to use two robots working in tandem in order to reduce bottlenecks and to increase productivity. This chapter deals with minimum - energy path planning for two robots working simultaneously in the same workspace.

Given the nature of the job to be performed by these two robots, say Robot 1 and Robot 2 , two different approaches are considered to solve the problem of minimizing the energy of each of these robots while simultaneously avoiding collision with the other. In the first case, it is assumed that Robot 1 has higher priority. This is true when there already exists a robot and another robot is introduced to aid this robot. In this case the problem is formulated to treat Robot 1 as a dynamic obstacle which has to be avoided when planning the motion of Robot 2.

In the second case, it is assumed that the initial and final points on the trajectories of Robot 1 ,as well as , Robot 2 along with the traversal time of each arm is specified. Given this information and the knowledge of actuator limits, paths for Robot 1 and Robot 2 are to be planned such that the overall expenditure of energy is

minimized and collision between the two arms is avoided.

3.1 Problem Formulation :

For simplicity two identical robots are considered. This , however , need not be a restriction and any two different types of robots can be considered.

The minimum - energy path planning problem is then formulated in two different ways. They are as defined below :

Problem Formulation 1 :

Find the minimum - energy path for Robot 1 assuming that Robot 2 is not present in the workspace. The optimal trajectory thus obtained is given by

$\underline{X}_1{}^* \in R^3$

and the optimal control is given by

$\underline{U}_1{}^* \in R^3$

The cost function assumed is of the form

$$J_1 = K_{11} \int_0^{t_{f1}} KE \, dt + K_{12} \int_0^{t_{f1}} PE \, dt \qquad (3.1a)$$

where K_{11} and K_{12} are constants and KE and PE the kinetic and potential energies of the robot. Then, given the initial and final configurations for Robot 2 and the time of traversal t_{f2} , find $\underline{X}_2{}^*$ and $\underline{U}_2{}^*$ by minimizing a cost function given by

$$J_2 = K_{21} \int_0^{t_{f1}} KE_2 \, dt + K_{22} \int_0^{t_{f2}} PE_2 \, dt \qquad (3.1b)$$

where K_{21} and K_{22} are constants and KE_2 and PE_2 are the

kinetic and potential energies respectively of Robot 2 under the following constraints

$$\underline{X}_2^* \cap \underline{X}_1^* = \emptyset \quad \text{for all } 0 < t < \max(t_{f1}, t_{f2}) \tag{3.2}$$

$$\underline{u}_2^* \in \underline{U}_2^* \tag{3.3}$$

To solve this problem, the method adopted in Chapter 2 is utilized. The state vector formulation is the same as eqn.(2.2) and eqn.(2.3) . The essential difference is that while collision checking in Chapter 2 was for static obstacles , here the collision checking is done for dynamic obstacles. After each variation of a component of the state vector, the new position of Robot 2 is calculated and knowing the position of Robot 1 at this instant, checks are made to find out if this new position of Robot 2 results in collision with Robot 1 at this instant.

Problem Formulation 2 :

Given the initial and final configurations of Robot 1 and Robot 2 , find $\underline{X}_i^*$ and $\underline{U}_i^*$, i=1,2 by minimizing the following cost function

$$J_{1,2} = J_1 + J_2 \tag{3.4}$$

subject to the constraints

$$\underline{X}_1^*(T) \cap \underline{X}_2^*(T) = \emptyset \qquad \text{for } 0 < t < \max(t_{f1}, t_{f2})$$

$$\underline{u}_i^*(t) \cap \underline{U}_i(t) \qquad i = 1,2$$

where the subscripts refer to the robot in consideration.

To solve this problem, first an extended state vector is formed as given below.

$$x1 = q_{11}$$
$$x2 = q_{21}$$
$$x3 = q_{31} \qquad (3.5)$$
$$x4 = q_{12}$$
$$x5 = q_{22}$$
$$x6 = q_{32}$$

where the first subscript of the terms on the R.H.S. refers to the joint and the second subscript refers to the robot in consideration.

The nominal trajectories for the two robots are constructed. The nominal trajectories must not violate any of the constraints on bounds or control inputs. Further, the trajectories chosen must be collision - free. Each of the trajectories is divided into 'N' equal subintervals. The subinterval costs are calculated and stored. The overall starting cost is found by eqn.(3.4).

The variation step size for each of the components of the state vector is chosen. The variation is then performed for all the six elements of the state vector at each of the N-1 instants. After each variation, tests are performed to check whether

(i) the states are admissible

(ii) the resulting position does not cause collision with the other robot.

(iii) the control inputs required to translate the states from $\underline{X}_i(k-1)$ to $\underline{X}_i(k+1)$ via $\underline{X}_i(k)$; (i=1,2) are admissible.

(iv) the sum of subinterval costs for the intervals k-1

to k+1 is less than the sum of the costs before the variation.

If the above tests give positive results then the original state at instant k is replaced by the perturbed value of the state. This process is repeated iteratively till no further reduction in costs are obtained. Then the perturbation step size is halved and the process repeated. The algorithm is terminated when satisfactory convergence is obtained.

3.2 Geometric constraint checking :

To simplify geometric constraint checking, certain assumptions about the robot arm are made. It is assumed that collisions can occur between the combination of the wrists and the third links of the manipulators. As stated before, the wrist is modeled as a sphere and the third link is modeled as a cylinder.

To determine whether a collision exists, it is only required to know the positions of the first three joints of each robot arm. The procedure for checking collisions is as follows.

Refer to Fig. 3.1 which depicts a typical arrangement of the two arms. Once the joint positions of the two arms are known, the coordinates of the points A,B,C and D can be calculated by forward kinematic relations.

Define the following quantities for the identical robots.

l_c :length of each cylinder.

r_c :radius of each cylinder.

r_s :radius of each sphere.

The following cases are considered for collision detection:

(i) collision between wrist of Robot 1 and wrist of Robot 2.

(ii) collision between wrist of Robot 2 and third link of Robot 1.

(iii) collision between wrist of Robot 1 and third link of Robot 2.

(iv) collision between third link of Robot 1 and third link of Robot 2.

They are discussed in detail below.

3.2.1 Collision between the wrist and wrist :

This is the simplest of the four cases. Let $B(x_2,y_2,z_2)$ and $C(x_3,y_3,z_3)$ be the centers of the wrists of robots 1 and 2 respectively. Then the distance BC is given as

$$BC = [\ (x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2\ ] \qquad (3.6)$$

If $BC \leq 2\ R_S$

then a collision between the two wrists is a definite occurrence.

3.2.2 Collision between wrist of Robot 2 and the third link of Robot 1 is checked as follows:

Let $A(x_1,y_1,z_1)$ and $B(x_2,y_2,z_2)$ represent the end points of the axis of the cylinder of Robot 1.

Similarly, $C(x_3,y_3,z_3)$ and $D(x_4,y_4,z_4)$ represent the corresponding points for Robot 2. Let l_1,m_1,n_1 and l_2,m_2,n_2 be the direction cosines of AB and CD respectively. Then,

$$l_1 = \frac{(x_2 - x_1)}{l_c}; \qquad m_1 = \frac{(y_2 - y_1)}{l_c}; \qquad n_1 = \frac{(z_2 - z_1)}{l_c};$$

$$l_2 = \frac{(x_4 - x_3)}{l_c}; \qquad m_2 = \frac{(y_4 - y_3)}{l_c}; \qquad n_2 = \frac{(z_4 - z_3)}{l_c}; \qquad (3.7)$$

First, the shortest distance between the sphere and the axis of the cylinder is to be computed. If this distance is less than or equal to $(r_s + r_c)$, and if the point of contact lies on the cylinder, then collision exists. Refer to Fig.3.2 where Fig.3.2a depicts the case where the shortest distance is less than or equal to $(r_c + r_s)$ and the point of contact lies on the cylinder. Here collision occurrence is positive. Fig.3.2b illustrates the case where 'C'is at a distance less than or equal to $(r_s + r_c)$ but the point of contact does not lie on the cylinder. Here, collision does not occur.

Let the equation of the line AB be

$$\frac{(x - x_1)}{l_1} = \frac{(y - y_1)}{m_1} = \frac{(z - z_1)}{n_1} = r \qquad (3.8)$$

where l_1,m_1,n_1 are calculated according to eqn. (3.7). From eqn.(3.8) any point on AB has to satisfy

$$\begin{aligned} x &= x_1 + l_1 \,.\, r \\ y &= y_1 + m_1 \,.\, r \qquad\qquad (3.9) \\ z &= z_1 + n_1 \,.\, r \end{aligned}$$

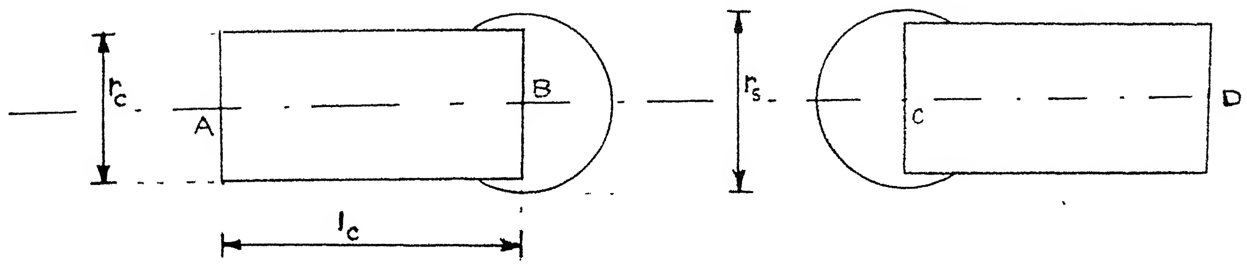


Fig. 3.1 : Typical arrangement of the arms.

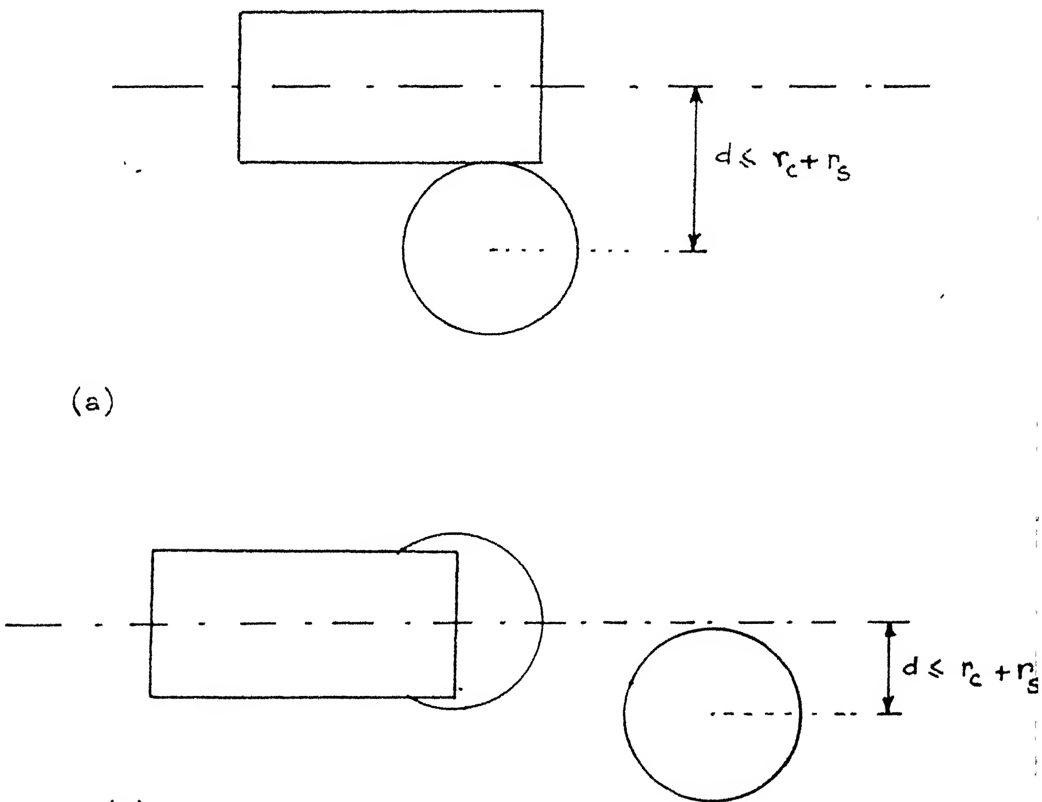


Fig. 3.2 : Collision check for wrist and third link.

Since the shortest distance, 'CE' is perpendicular to AB,

$$l\, l_1 + m\, m_1 + n\, n_1 = 0 \qquad (3.10)$$

Where l,m,n are the direction cosines of CE. From eqn.(3.9) and (3.10) and since

$$l_1^2 + m_1^2 + n_1^2 = 1$$

the value of 'r' is

$$r = l_1(x_3-x_1) + m_1(y_3-y_1) + n_1(z_3-z_1) \qquad (3.11)$$

knowing r, the coordinates of E can be calculated from eqn.(3.9). Now the length of the straight line joining points C and E is

$$|CE| = \sqrt{[\,(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2\,]} \qquad (3.12)$$

If CE $\leq (r_s + r_c)$, then a potential for collision exists and further checks are needed to test for positive occurrence of collision. To do that, join point C to points A and B and find the projection of AC and BC on AB. (similar to section 2.2). These projections are given by

$$\text{Proj. } AC = AE = ABS[(x_3 - x_1)l_1 + (y_3 - y_1)m_1 + (z_3 - z_1)n_1] \qquad (3.13)$$

$$\text{Proj. } BC = BE = ABS[(X_3 - X_2)l_1 + (Y_3 - Y_2)m_1 + (z_3 - z_2)n_1] \qquad (3.14)$$

If $(AE + BE) \leq AB + 2\, r_s$, then collision will definitely occur.

The collision between the wrist of Robot 1 and third link of Robot 2 is checked in a similar fashion.

3.2.3 Collision between the third links of the two robots is checked as follows :

Let P be any point on the straight line CD at a distance of 2 r_c from AB. Since P lies on CD the following equations are satisfied.

$$x = x_3 + l_2\ r$$
$$y = y_3 + m_2\ r \qquad (3.15)$$
$$z = z_3 + n_2\ r$$

where (x,y,z) are the coordinates of P.

Refer to Fig.3.3 . A collision will occur if the following hold:

(i) the point P lies on the segment CD and

(ii) the foot of the perpendicular from P onto the straight line AB lies on the segment AB.

Let

AM = Projection of AP on AB

BM = Projection of BP on AB.

The distances AM and BM are given by

$$|AM| = ABS[(x-x_1)l_1 + (y-y_1)m_1 + (z-z_1)n_1]$$
$$\qquad (3.16)$$
$$|BM| = ABS[(x-x_2)l_1 + (y-y_2)m_1 + (z-z_2)n_1]$$

From eqns.3.15 and 3.16

$$AM^2 = T1^2 + 2\ T1\ T2\ r + T2^2\ r^2$$
$$\qquad (3.17)$$
$$AP^2 = T3 + T4\ r + r^2$$

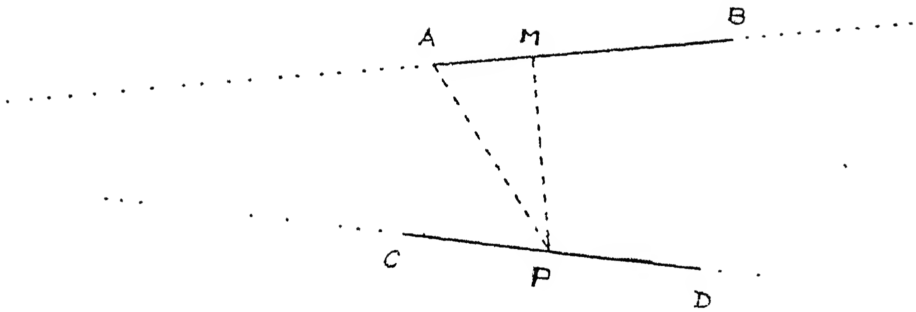


Fig. 3.3 : Collision check for the third links.

where

$$T1 = (x_3 - x_1)l_1 + (y_3 - y_1)m_1 + (z_3 - z_1)n_1$$

$$T2 = l_1 l_2 + m_1 m_2 + n_1 n_2$$

$$T3 = (x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1) \tag{3.18}$$

$$T4 = 2[\ (x_3 - x_1)l_2 + (y_3 - y_1)m_2 + (z_3 - z_1)n_2\ ]$$

Again from triangle APM, using Pythagoros Theorem,

$$PM^2 = AP^2 - AM^2$$
$$=[\ T3 + T4\ r + r^2] - [T1^2 + 2\ T1\ T2\ r + T2^2\ r^2\ ] \tag{3.19}$$

The next step is to find P such that PM = 2 r_c; i.e. the point at which the two cylinders touch each other. Solving eqn.(3.19), the value of r which makes this feasible can be determined. Since eqn.(3.19) is a quadratic in r , a real value of r exists only if the discriminant is greater than or equals zero. If a real value of r exists, then a potential for collision exists and further checks are necessary to determine positive occurrence.

Let r_1 and r_2 be the two real roots of r. Then for each r_i, i=1,2 find the points P and M. Collision occurs if both P and M lie within CD and AB respectively.

3.3 Numerical results :

The MLV algorithm is applied to the problem formulated in 3.1 and the results verified through digital computer simulations. The efficacy of the collision checking scheme is clearly brought out through the following examples.

Example 3.1 (Problem formulation 1):

In this example Robot 1 is required to move from (0.5,0.2,1.2) to (0.6,-0.2,1.1) in the Cartesian coordinate space and Robot 2 is required to move from (0.5,-0.2,1.0) to (0.6,0.2,1.2) simultaneously. The traversal time for both the robots is 1 second. Constant bounds on the input torques of both the robots are assumed. They are as given in Example 2.1 . Robot 1 is assumed to have higher priority.

Initially, Robot 1 path is planned using MLV algorithm with K_{11} and K_{12} chosen as in Example 2.1. i.e. 15 and 0.5 respectively. The minimum cost for this path is found to be 76.48195 units.

Given this path, a minimum-energy path for Robot 2 is sought. An initial feasible trajectory is chosen which avoids colliding with with Robot 1 traveling in its optimum path. This path is such that its projection on X-Y plane is a semicircle and motion in the vertical plane is in equal increments. The cost function chosen is identical to that of Robot 1.

The initial cost along the nominal trajectory of Robot 2 is 142.4389 units. On application of MLV this was subsequently reduced to 108.2730 units. The optimal total cost of the two robots is 184.7549 units. The X-Y plots for the initial and final trajectories of the two robots are shown in Figs.3.4 and 3.5 respectively.

Example 3.2 (Problem formulation 2) :

In this example, the initial and final configurations for both the robots are the same as in example 3.1 . The cost function in this case is given by eqn.(3.4) with K_{11}, K_{12} , K_{21} and K_{22} as in example 3.1 .

The initial feasible trajectories for Robot 1 and Robot 2 have semicircular projections on X-Y plane and equal increments / decrements in the vertical plane. The initial cost was 119.4910 units for Robot 1 and 142.4389 units for Robot 2. The overall cost being 262.9299 units.

The MLV algorithm was applied according to problem formulation 2. The minimized cost for Robot 1 was 83.1579 units and for Robot 2 was 95.5453 units. The overall minimized cost was found to be 180.7032 units. The optimal paths for the two robots according to this problem formulation is shown in Fig.3.6 . The joint position plots of the two robots for the nominal trajectory, as well as, for examples 3.1 and 3.2 are shown in Figs. 3.7 - 3.12 .

3.4 Conclusions :

In this chapter, a method for checking and avoiding collisions with dynamic objects in the work space of the robot has been presented. This method has been adapted for planning collision-free paths for two robots. The efficacy of collision avoidance scheme is clearly brought out in Figs.3.5 and 3.6 .

It is found that the overall cost resulting from problem formulation 2 is not much lower compared to the

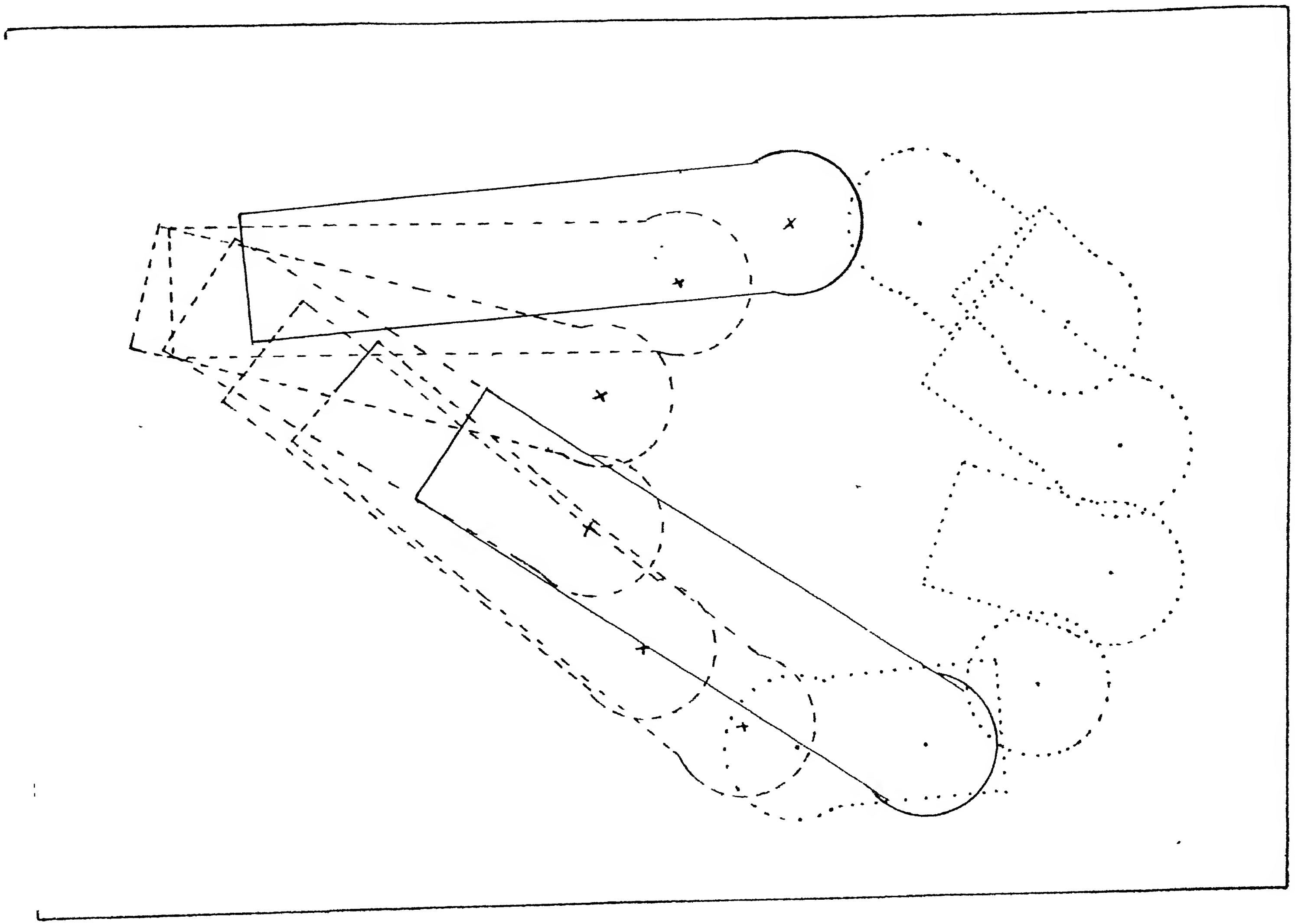


Fig. 3.4 : X-Y plot of the nominal trajectories of the two arms.

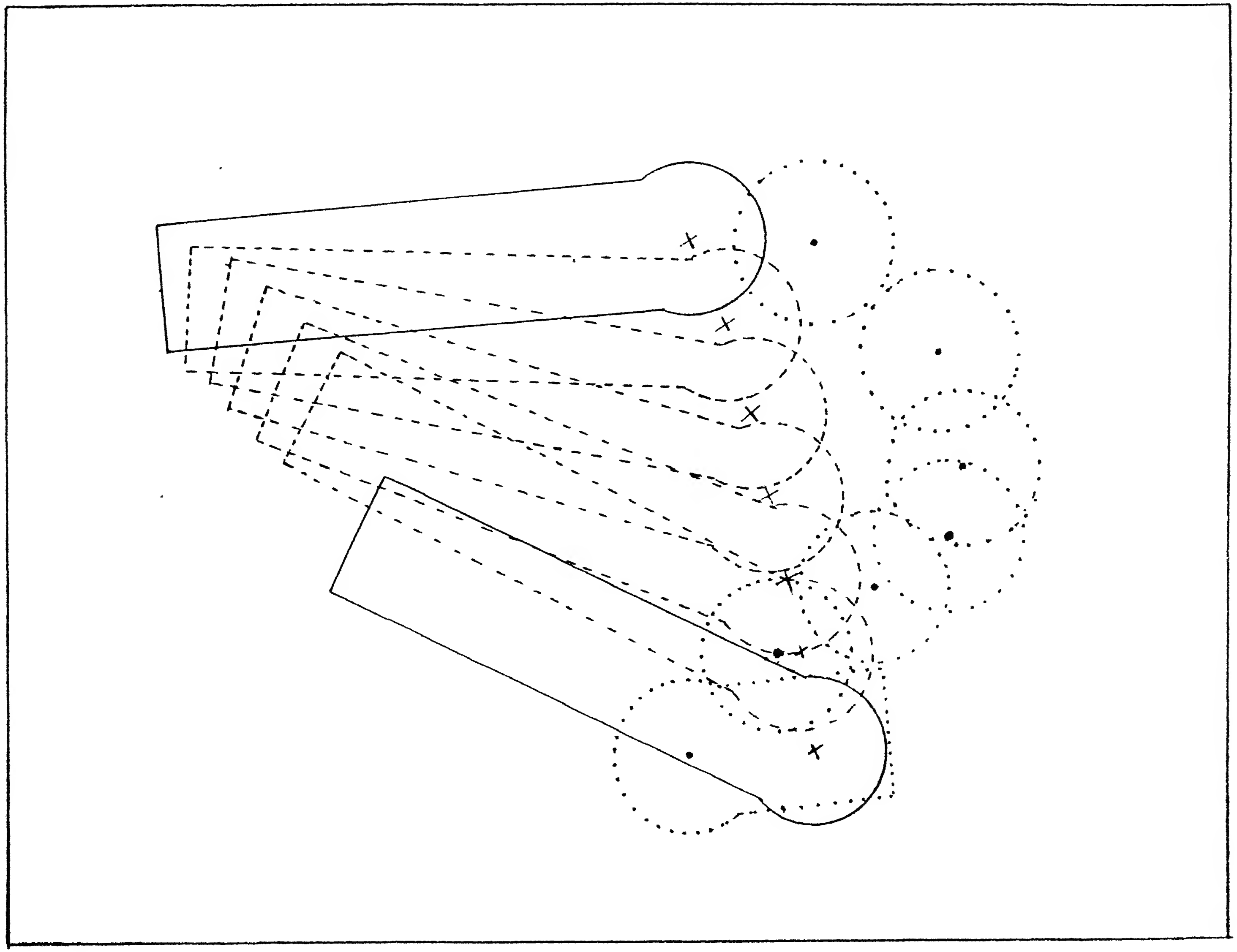


Fig. 3.5 : X-Y plot of the optimal trajectories of the two arms Ex. 3.1.

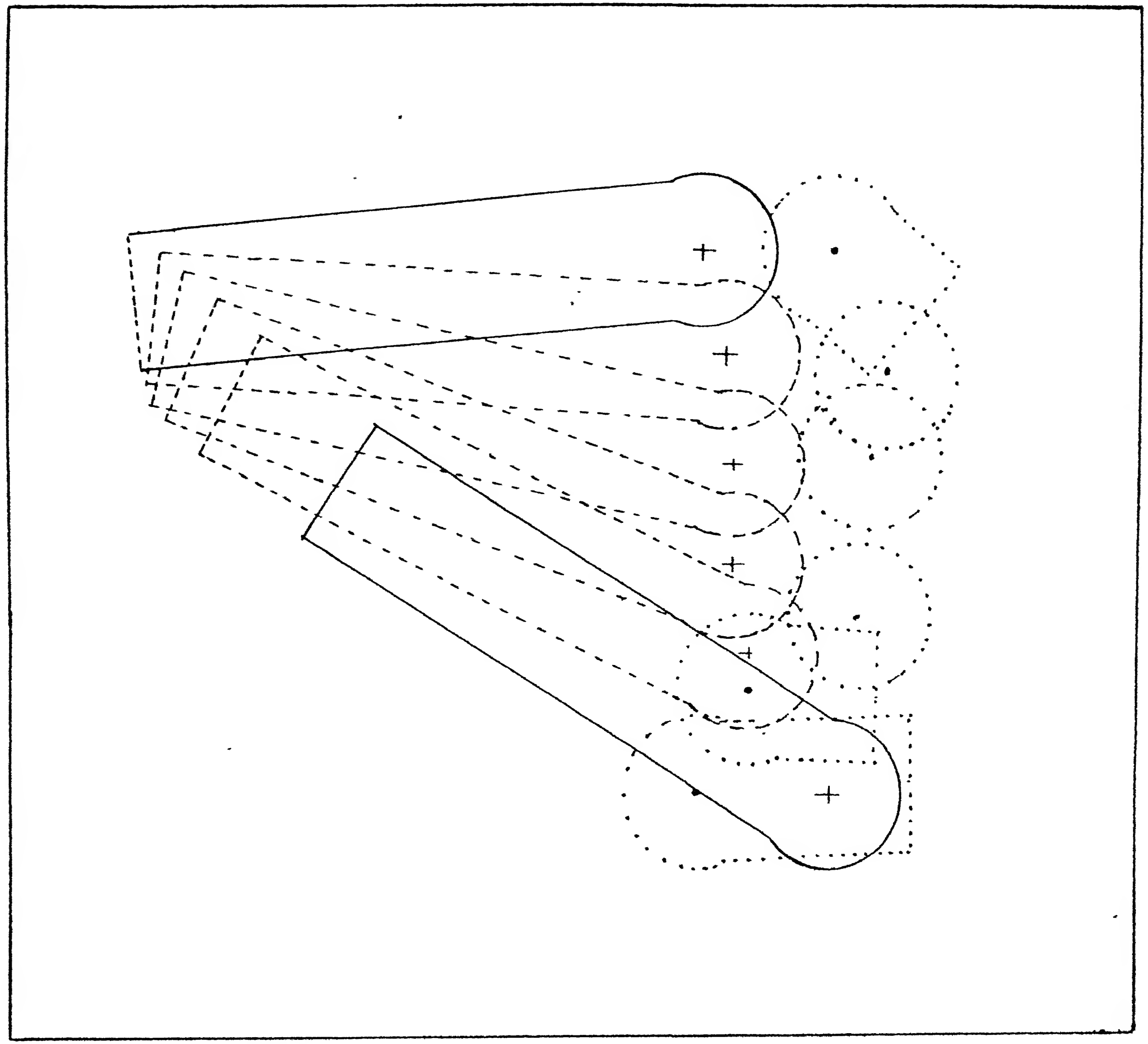


Fig. 3.6 : X-Y plot of the optimal trajectories of the two arms Ex. 3.2.

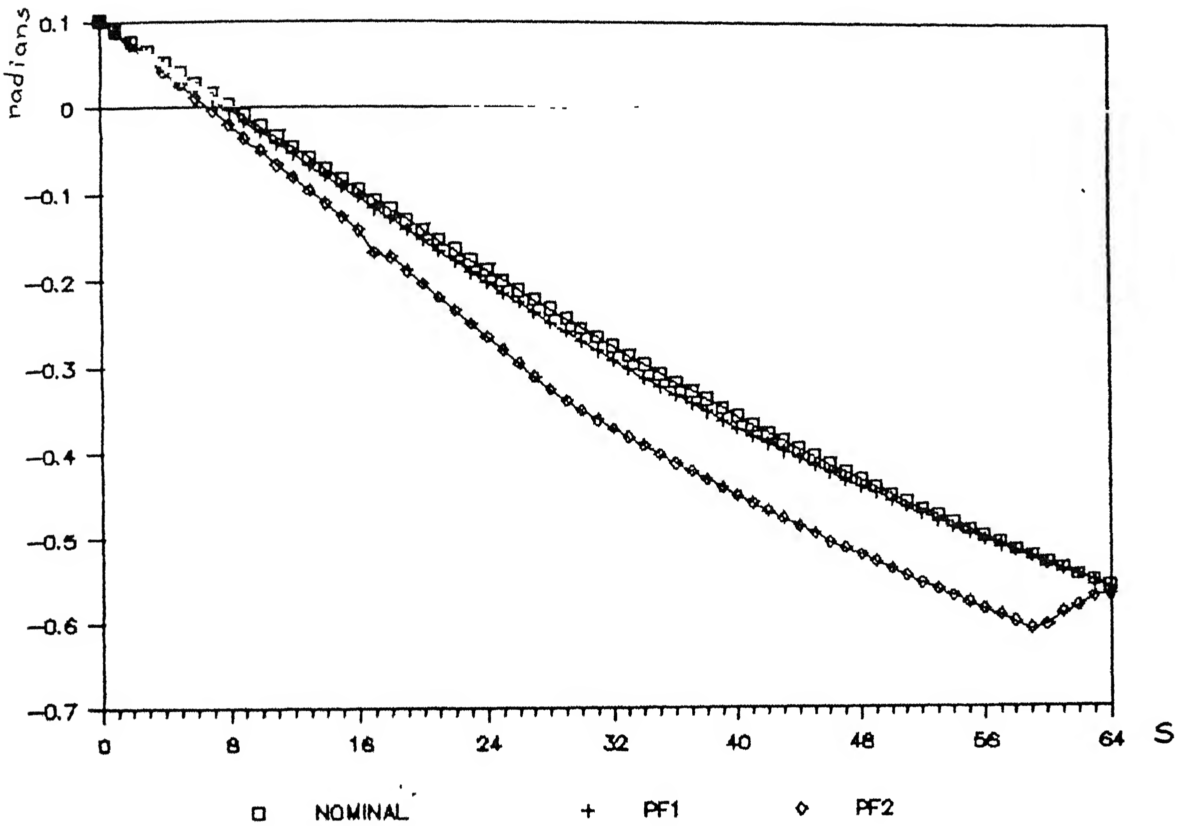


Fig. 3.7 : Joint 1 position plots for Robot 1.

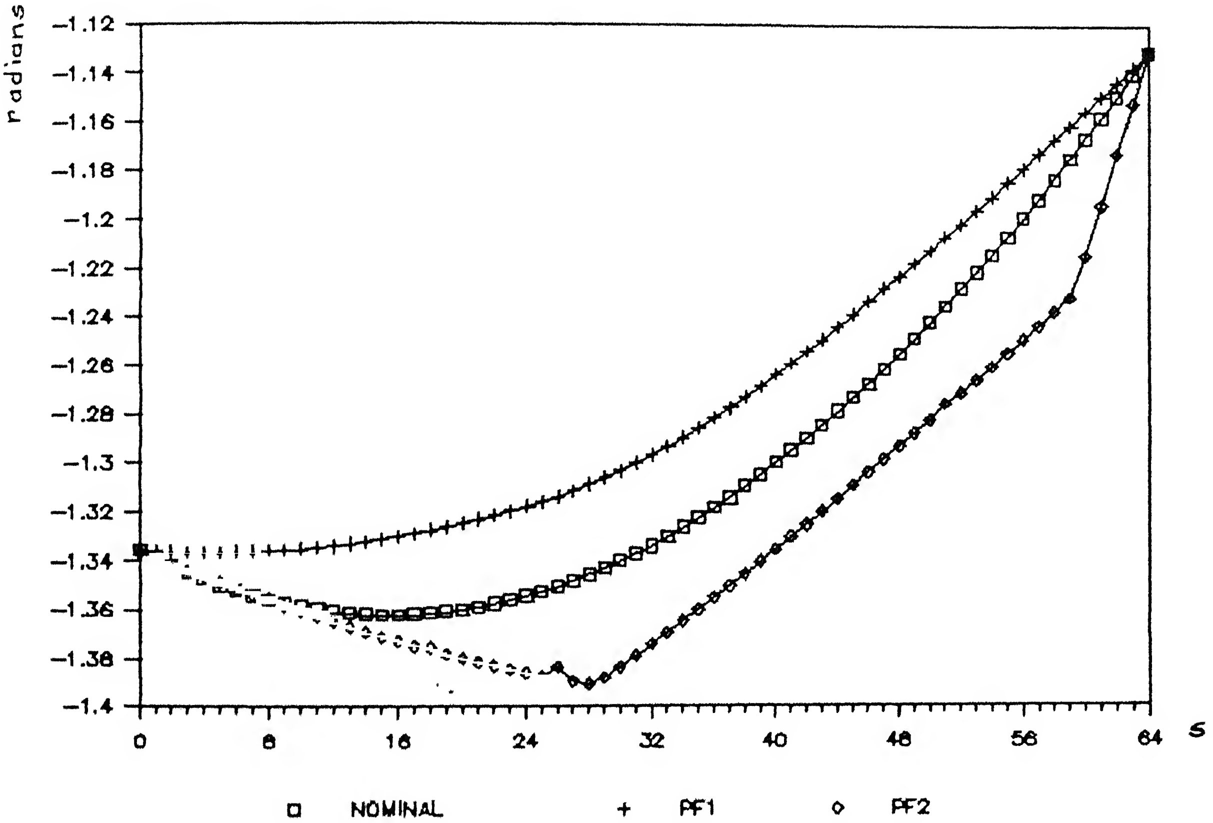


Fig. 3.8 : Joint 2 position plots for Robot 1.

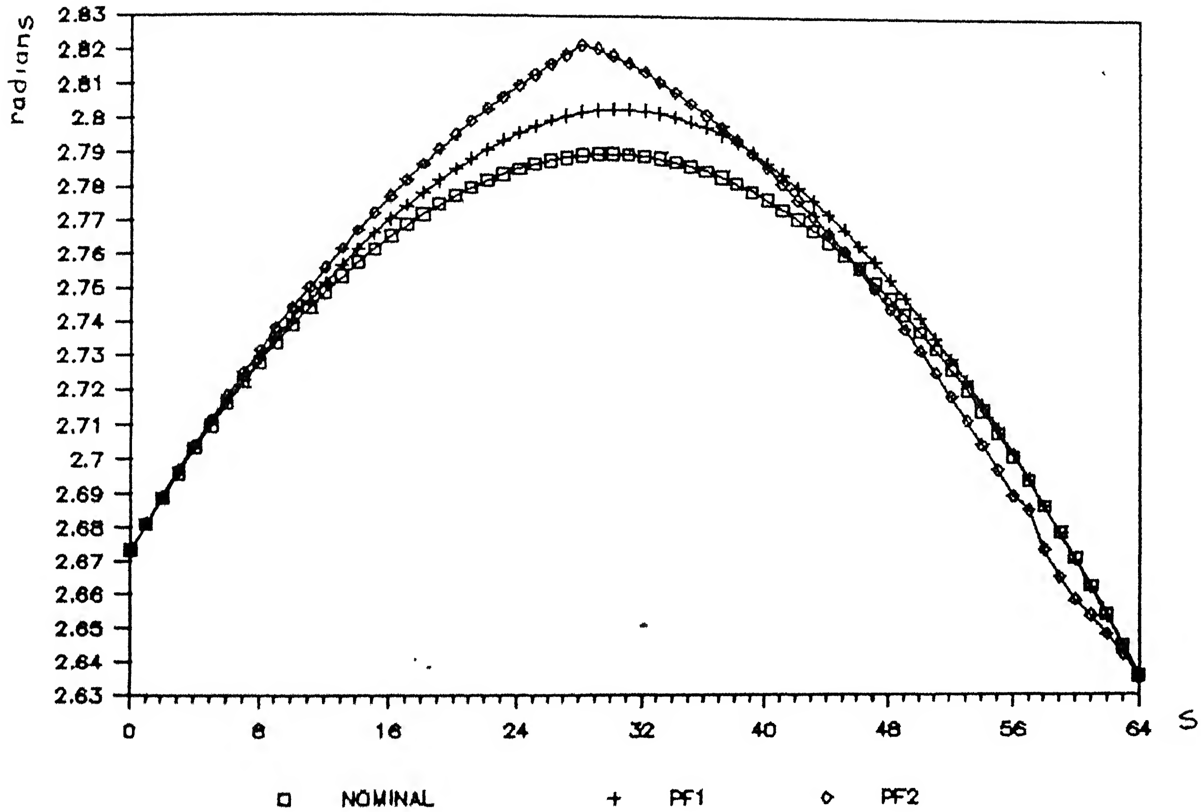


Fig. 3.9 : Joint 3 position plots for Robot 1.

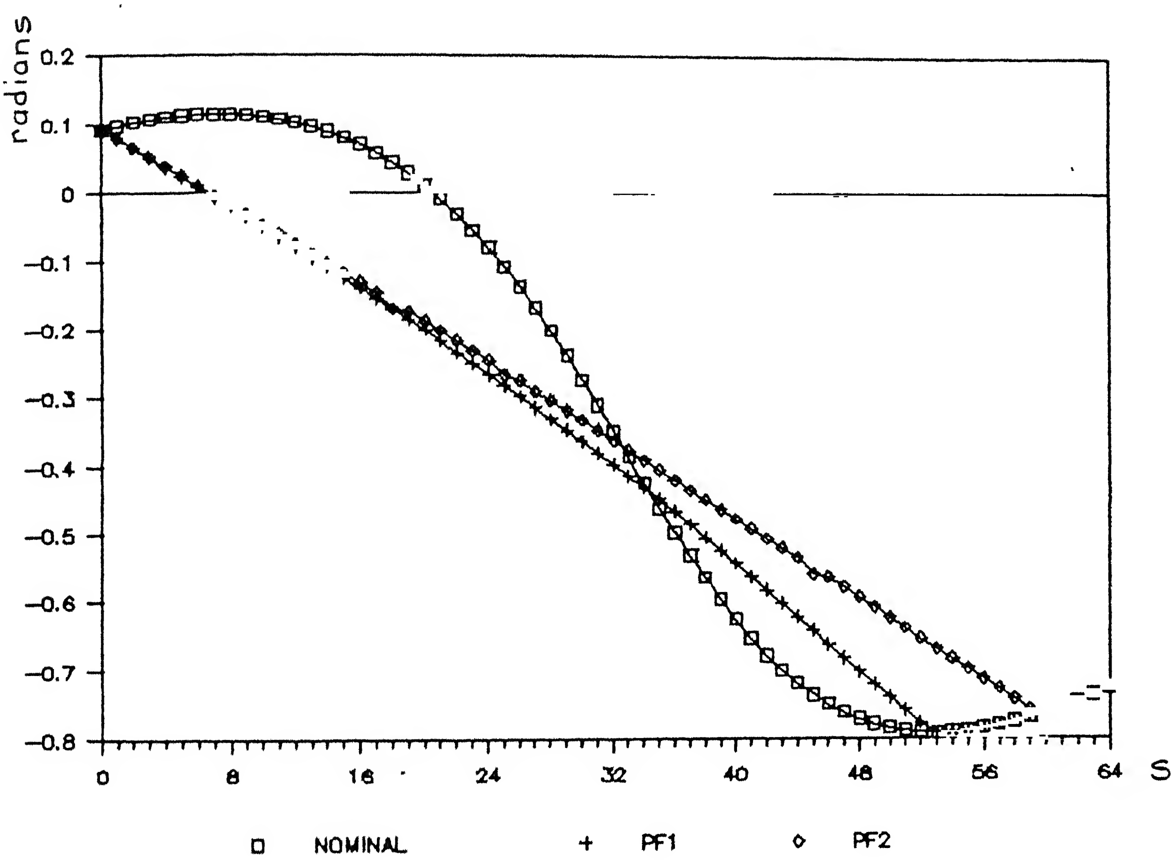


Fig. 3.10 : Joint 1 position plots for Robot 2.

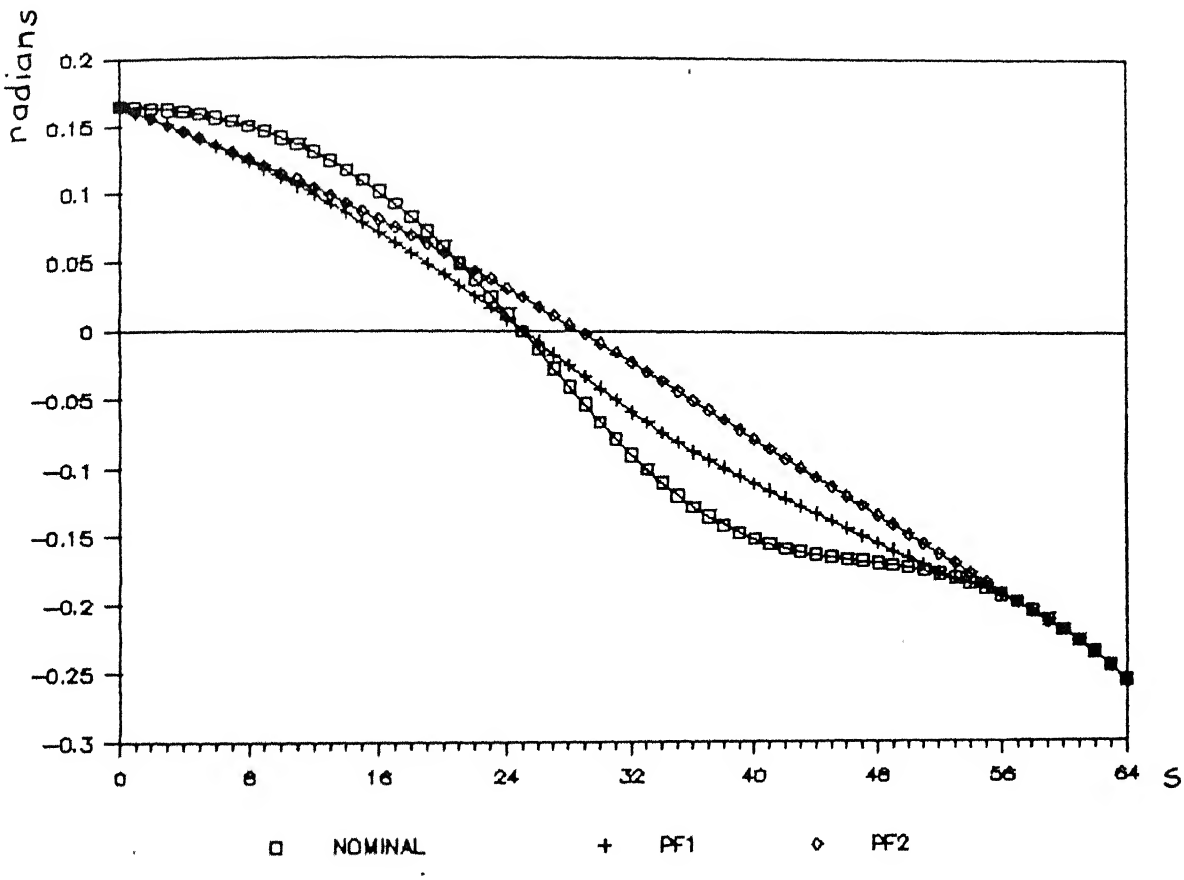


Fig. 3.11 : Joint 2 position plots for Robot 2.

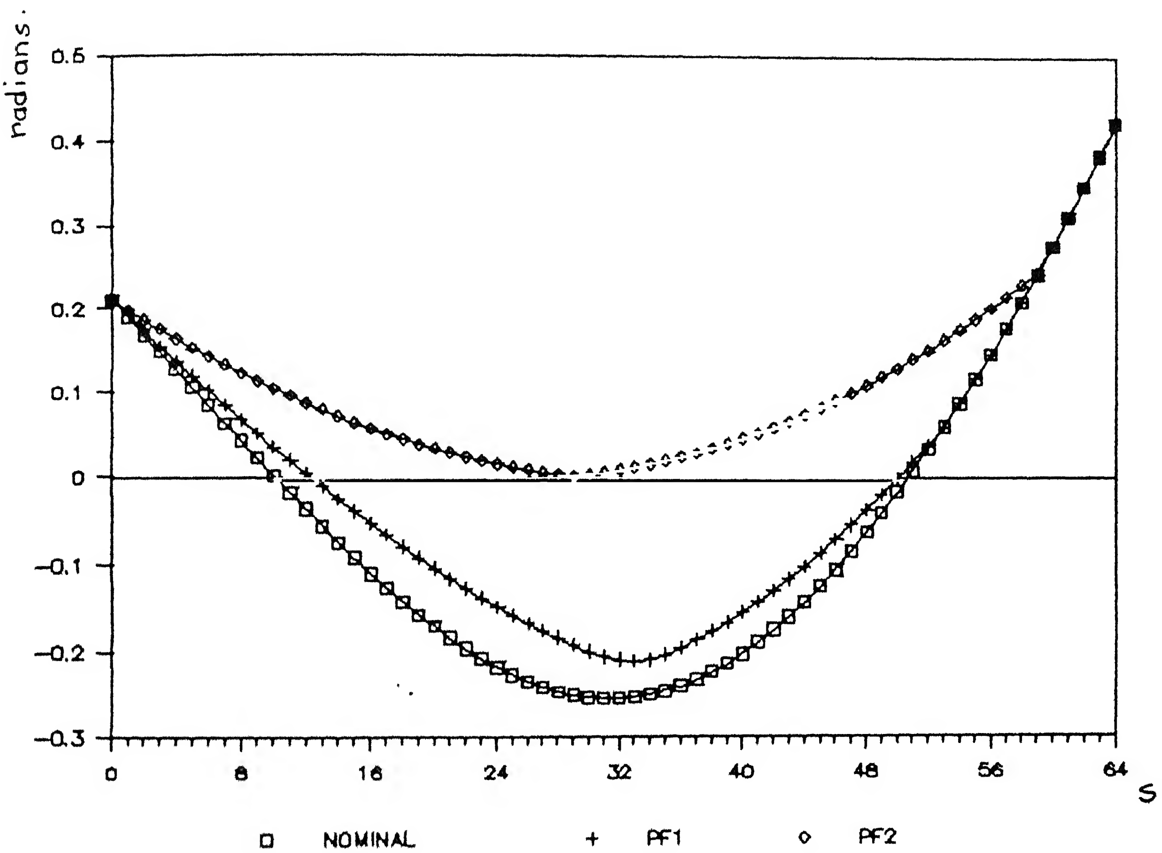


Fig. 3.12 : Joint 3 position plots for Robot 2.

cost resulting from problem formulation 1. This is because the cost for Robot 1 in example 2 is greater than the cost in example 1, as it is not traveling along the optimum path in this case. This increase in cost of Robot 1 partly offsets the reduction in cost of Robot 2 obtained in problem formulation 2. Thus, the above two formulations can be used to determine if it is necessary to replan the path for both the robots, when a robot is introduced into the work space of an already existing robot.

CHAPTER 4.

COLLISION-FREE MINIMUM-TIME PATH FOR TWO ROBOTS.

In this chapter, collision - free minimum-time trajectory planning for two robots using the method of Local Variations is presented. The problem formulations assume that the paths to be followed by the end - effectors of the two robots are specified 'a priori '. The traversal time for each robot is to be minimised, while simultaneously avoiding collision with the other robot.

In the minimum - energy path planning problems of Chapters 2 and 3, the time of traversal was a fixed parameter. In the minimum - time path planning problem however, this is not the case, and the optimal time is to be calculated. Hence, the strategies of chapter 2 and 3 cannot be applied directly. To overcome this problem, the parameterization approach is used [23,25].

The path specified in the Cartesian coordinate space is first converted to a path in joint coordinate space and is then expressed in terms of a single parameter. Collision checking scheme developed in Chapter 3 is used to check for collisions. An optimal solution is then obtained by applying the MLV algorithm in phase plane. The formulation is similar to that of Shin and McKay [25]. The only difference between the two methods being in the optimization algorithm used. Shin and McKay used dynamic programming as an optimization technique in phase plane.

However, when MLV algorithm is used , the method becomes much simpler to apply, as well as, reduces computation and memory requirements [26].

4.1 Problem formulation :

First the parameterization approach for a single robot is discussed. It is then extended for two robot case.

In the minimum - time path planning problems, the cost function is expressed as

$$T = \int_{0}^{tf} dt \qquad (4.1)$$

where the final time t_f is left free. Because of this reason time can no longer be treated as a stage variable. The parameterization approach has a dual advantage. It not only takes care of the above mentioned difficulty but also reduces the dimensionality of the problem from 'n' for a n joint manipulator to one.

The parameterization is achieved by expressing the joint positions as functions of a single parameter 's'. It is assumed that the path of the robot does not retrace as s varies from zero to s_{max}.

Thus

$$q_i(s) = f_i(s) \qquad 0 \le s \le s_{max} \quad \text{for } i = 1,..,n \qquad (4.2)$$

where the q_i's represent the joint positions of the robot along the path. The dynamic equations of motion of a 'n '

joint manipulator given by eqn.(2.3) is reproduced here.

$$\tau_i = \sum_{j=1}^{n} D_{ij} \ddot{q}_j + \sum_{j=1}^{n} \sum_{k=1}^{n} H_{ijk} \dot{q}_j \dot{q}_k + G_i \qquad i = 1,..,n$$

Defining

$$\mu = \frac{ds}{dt}$$

and expressing q, $\dot{q}$, $\ddot{q}$ in terms of 's' , the following equations are obtained.

$$\dot{q}_i = \mu \frac{df_i}{ds}$$

$$\ddot{q}_i = \mu^2 \frac{d^2 f_i}{ds^2} + \dot{\mu} \frac{df_i}{ds} \tag{4.3}$$

Therefore, the parameterized dynamics equation can be written as

$$\mu_i = M_i \dot{\mu} + Q_i \mu^2 + G_i \tag{4.4}$$

where $M_i = \sum_{j=1}^{n} D_{ij} \frac{df_j}{ds}$

and

$$Q_i = \sum_{j=1}^{n} D_{ij} \frac{d^2 f_j}{ds^2} + \sum_{j=1}^{n} \sum_{k=1}^{n} H_{ijk} \frac{df_j}{ds} \frac{df_k}{ds} \tag{4.5}$$

Given the paths of both the robots in joint space, each robot's path and dynamics can be expressed in terms of a single parameter. If the motion of Robot 1 is characterized by a parameter s_1 and the motion of Robot 2 is characterized

by parameter s_2, then the state equations can be formulated as

$$\dot{s}_1 = \mu_1$$
$$\dot{s}_2 = \mu_2 \qquad (4.6)$$

$$u_{i1} = M_{i1}\,\mu_1 + Q_{i1}\,\mu^2 + G_{i1}$$
$$u_{i2} = M_{i2}\,\mu_2 + Q_{i2}\,\mu^2 + G_{i2} \qquad (4.7)$$

for $i = 1, \ldots, n$

where the second subscript gives the robot number.

μ, M, Q and G are as defined in eqns.(4.3) - (4.5).

The time required to move the manipulators from the initial configuration to final configuration is given by

$$T1 = \int_0^{tf_1} dt = \int_0^{tf_1} \frac{dt}{ds_1}\,ds_1 = \int_0^{tf_1} \frac{1}{\mu_1}\,ds_1 \qquad (4.8)$$

$$T2 = \int_0^{tf_2} \frac{1}{\mu_2}\,ds_2 \qquad (4.9)$$

where T1 and T2 are the traversal times of Robot 1 and Robot 2, respectively.The overall cost is then given by

$$T = K1*T1 + K2*T2 \qquad (4.10)$$

where K1 and K2 are constants chosen depending on the relative weightings given to costs of Robot 1 and Robot 2.

The bounds on torques are arbitrary functions of positions and velocity and are expressed as

$$\underline{u}_{1,2}^{min}(s_{1,2},\mu_{1,2}) \le \underline{u}_{1,2} \le \underline{u}_{1,2}^{max}(s_{1,2},\mu_{1,2}) \qquad (4.11)$$

where $\underline{u} = [u_1, \ldots, u_n]^T$

If X1(t) denotes the set of all points on the

Robot 1 at time T, and X2(t) denotes the set of all points on Robot 2 at the same instant, then the positional constraint can be stated as

$$X1(t) \cap X2(t) = \emptyset \text{ for all } 0 \le t \le \max(T1,T2) \qquad (4.12)$$

The optimal control problem is to find $\underline{u}_{1,2}^*$ and $\mu_{1,2}^*$ so that T is minimum subject to the boundary conditions

$$\mu_{1,2}(0) = \mu_{1,2}^0 \quad \text{and}$$
$$\mu_{1,2}(s_{max}) = \mu_{1,2}^f \qquad (4.13)$$

and the constraints specified by eqns.(4.11) and (4.12)

Optimal control $\underline{u}_{1,2}^*$ can be found by minimizing the expression for T given by eqn.(4.7). Time can be minimized if the two robots travel at maximum acceleration or maximum deceleration. The μ - s plane being the phase plane, the optimal solution is a switching curve in this plane.

The variable μ is directly proportional to the velocities of the joints. Hence,it is called Psuedo-velocity. The traversal time for each robot can be minimized if all the joints travel at their maximum velocities. This means that the traversal time can be minimized if the Psuedo-velocity is maximized.

4.2 Solution to the Minimum - time problem using MLV :

Given the paths to be followed by the two robots, the range [0 to s_{1max}] and [0 to s_{2max}] is divided into N equal subintervals. The joint positions of the two

robots are expressed in terms of the parameters s_1 and s_2 as given in eqn.(4.2). The functions $f_{1,2}(s)$ need not be continuous functions of s_1 or s_2. It is sufficient if they are defined at each of these discrete points $s_{1,2}^{k}$, (k=1,..,n). The quantities $\frac{df_i}{ds_{1,2}}$ and $\frac{d^2 f_i}{ds_{1,2}^2}$ at each 'k' is evaluated and stored. Since the joint positions are known at discete points on the path, the coefficients of the dynamical equations (i.e. D_{ij}, H_{ijk}, G_i) at each of these points are calculated. Using these values and eqn.(4.5) the parameterized robot dynamic coefficients ($M_{i1}, M_{i2}, Q_{i1}, Q_{i2}, G_{i1}$ and G_{i2} ; i =1,..,3) at each of these points for the two robots are calculated and stored.

Nominal starting μ - s curves for the two robots are selected. The curves must satisfy the boundary conditions stated in eqn.(4.13). The nominal μ - s curves chosen must not violate the torque constraints or result in collision between the arms. The incremental costs associated with each subinterval for both the robots is calculated and stored. The total cost for each robot is the sum of all the incremental costs for its path. The overall cost T is calculated as given by eqn.(4.10). The incremental cost for interval [k-1 to k] is the time taken by the arm to move from configuration specified at k-1 to the configuration specified at k. It is calculated as

$$t_1 = \int_{s_{1,k-1}}^{s_{1,k}} \frac{1}{\mu_1} \, ds_1$$

$$t_2 = \int_{s_{2,k-1}}^{s_{2,k}} \frac{1}{\mu_2} \, ds_2 \qquad (4.14)$$

These integrals can be approximated as

$$t_1 = 2 \; s_1 \; / \; (\; \mu_1(k) - \mu_1(k-1) \;)$$

$$t_2 = 2 \; s_2 \; / \; (\; \mu_2(k) - B_2(k-1) \;) \qquad (4.15)$$

Variations are performed on each component of the state vector at each of the N-1 time instants. After each variation, tests are performed to check if any reduction in cost for the subintervals [k-1 ,k+1] has resulted. If reduction in cost is obtained, then further tests to check for torque violation and collisions are carried out. If the tests give negative results, then the state at this instant is replaced by the perturbed state. The algorithm is terminated when satisfactory convergence is reached [26].

4.3 Collision checking scheme:

Starting with two nominal trajectories for the two robots which are collision-free, does not ensure that the optimal path will remain collision-free if the MLV algorithm is applied to each path independently. This can be clearly brought out by the following example.

Example 4.1 Consider two robots whose paths are such that they intersect at some point X in the cartesian space (as shown in Fig.4.1). The robots move along this path but each robot passes through X at a different time thus avoiding collision. Consider the time instant 'T'when Robot 1 is at 'P'. After time δT Robot 1 passes through X while Robot 2 is at point 'R'. At time 'T+δT' they are still separated by sufficient distance to prevent collision . If Robot 2 occupies point X at a time 'T+ δT' ', Robot 1 would have moved to point 'V' in this time. Thus ,the nominal trajectories and the traversal time chosen result in a collision-free path.

If,the time of traversal for each robot is scaled by the same factor ,then the collision-free nature of the trajectories remain invariant. But in general,the nature of the paths being traversed, as well as the payload of individual Robots, make this impossible.

Consider the case where ,an application of MLV, there is no change in the path history of Robot 1 while Robot 2 is speeded up so that it passes through X at a time 'T+δT'. This results in collision at point X although the variation has occured at 's' corresponding to point R . Thus, it is not sufficient to check for collision only at the instant where the variation has occured, but it is necessary to check for collisions till the time both the robots reach their final configurations.

4.4 Numerical example :

Consider the problem where the end - effector of Robot 1 is required to move from a point (0.5,0.5,1.0) to a point (0.8,-0.2,0.8) in the Cartesian coordinate space. Simultaneously, Robot 2 is required to move from a point (0.72,-0.2,1.0) to a point (0.9,0.42,0.9). Robot 2 is positioned at a distance 1.47 m along the X-axis of Robot 1. It is desired to minimize the traversal time for both the robots, while avoiding collision between the two arms.

The nominal trajectories of both the robots are such that their projections on the X-Y plane are semi-circles, and motion in vertical plane is in equal increments / decrements. The nominal traversal time for each robot is taken as 1 second. Thus the overall cost is 2 units. The torque bounds are as in example 2.1. The collision checking scheme developed in section 3.2 is used to check for collisions between the two arms.

The nominal joint trajectories of both the robots are parameterized. A nominal μ - s curve is chosen for both the robots. For convenience, the trajectories of both the robots are divided into 64 equal subintervals. Hence the time for each subinterval for both the robots is 0.015625 seconds and 'μ' is 64 in all the subintervals.

The problem is formulated as explained in sections 4.1 & 4.2 and the MLV algorithm applied. The optimal time for Robot 1 was 0.48854 seconds and for Robot 2

was 0.32229 seconds. The overall cost being 0.81083 units. Now the minimum time along the nominal trajectory is found for each robot when the other robot is not present. In this case the minimum time for Robot 1 was found to be 0.25736 seconds and for Robot 2 the minimum time obtained was 0.21377 seconds.

The optimal μ - s curves for both the robots are shown in Figs. 4.2 and 4.3 . The torque plots for the first joint of both the robots are shown in Figs. 4.4 to 4.7. The three dimensional view of the motion of the two robots along the trajectory for both the nominal and optimal case are shown in Figs. 4.8 and 4.9 respectively.

4.4 Conclusions :

In this chapter, a method to minimize the traversal time for two robots working simultaneously in a workspace was presented. The collision checking scheme developed in Chapter 3 was incorporated in the MLV algorithm and the parameterization approach utilized to solve this problem.

From the optimal 'μ - s' curves, it is seen that the psuedo-velocity (μ) for both the robots is high leading to reduction in traversal time. However, in the initial part of the trajectory the value of psuedo-velocity is lesser than the corresponding value if only one robot was functioning. Further, the figures clearly indicate that the value of psuedo-velocity is the same in both the cases, with and without obstacle, once the two arms cross each other.

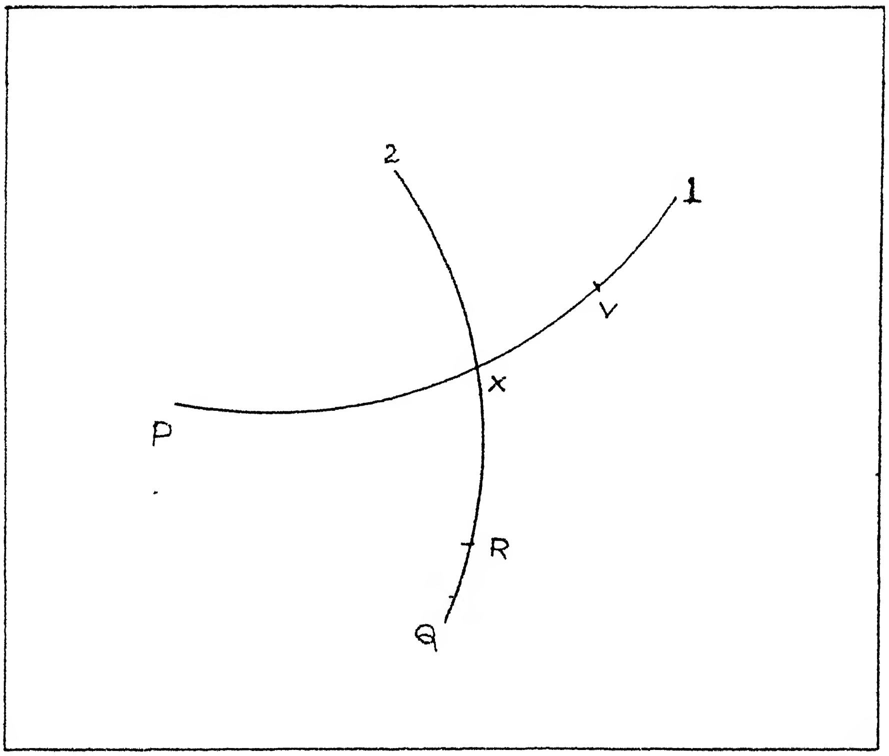


Fig. 4.1 : Intersecting paths of the two robots.

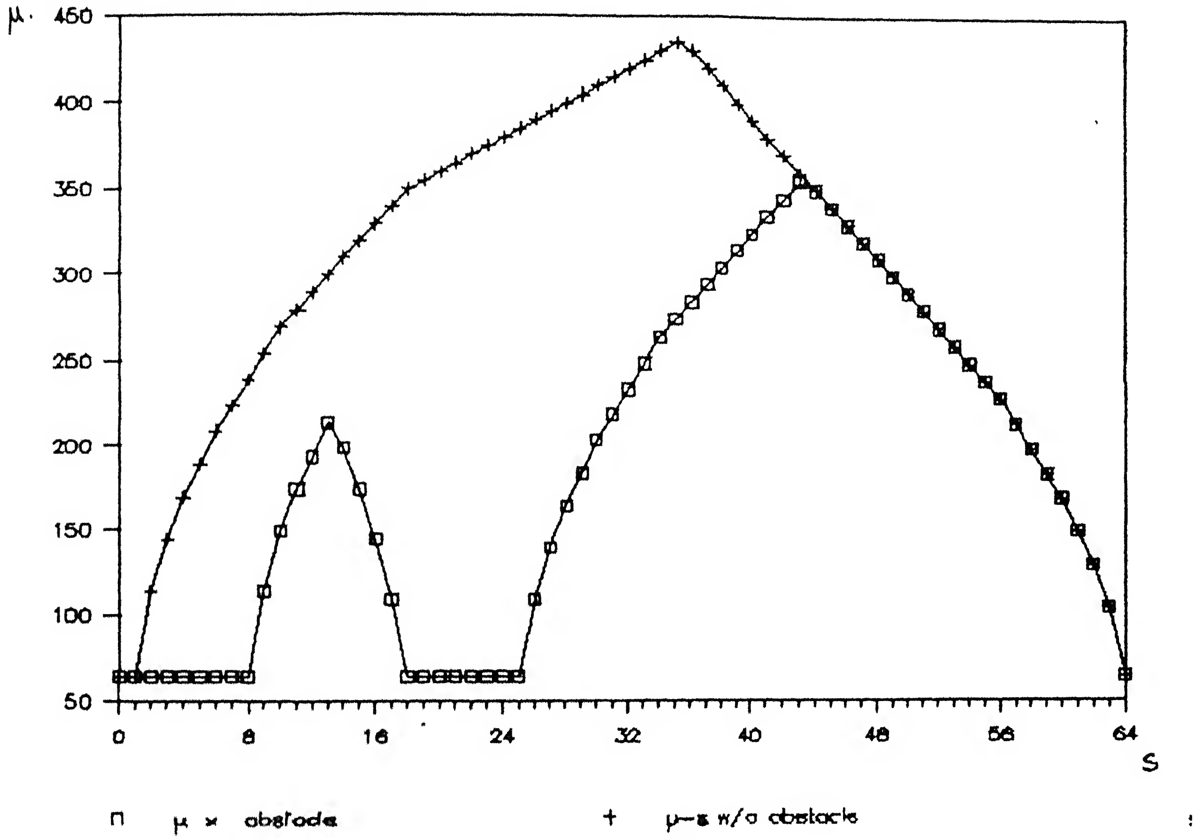


Fig. 4.2 : Phase-plane plot of Robot 1.

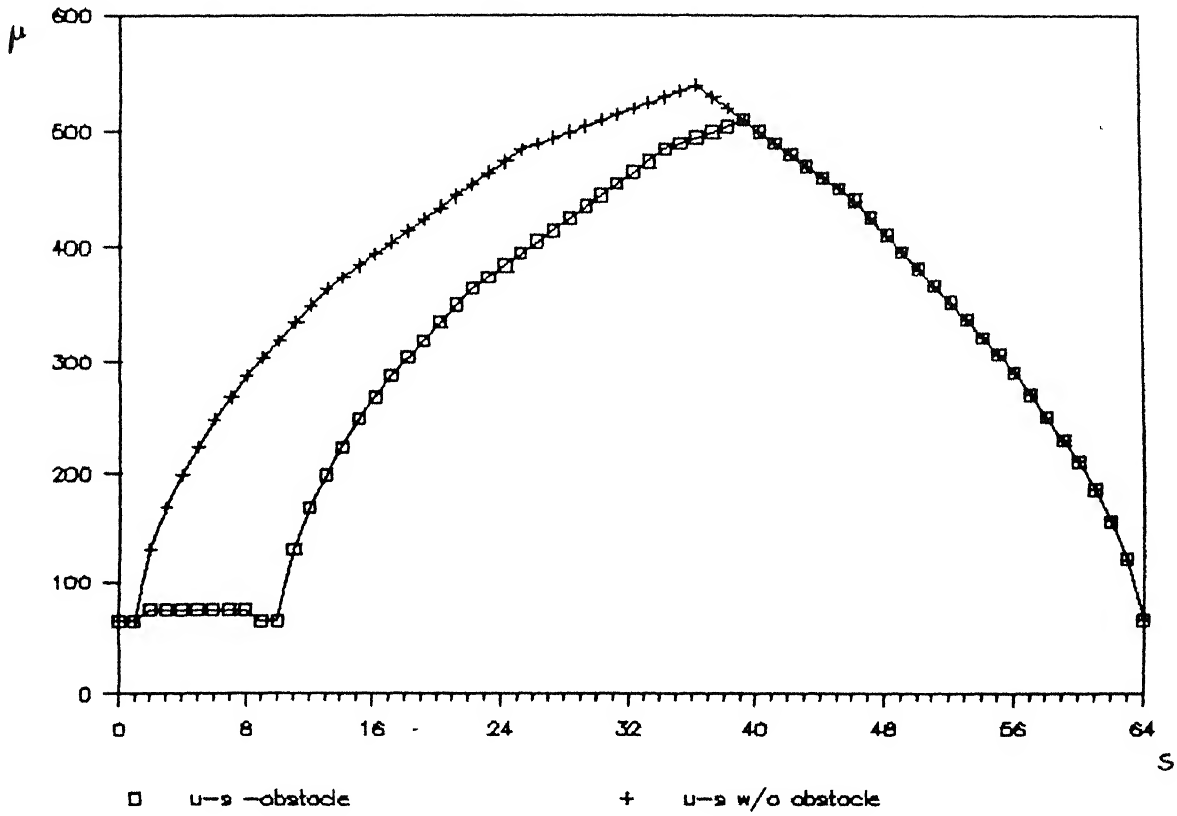


Fig. 4.3 : Phase-plane plot of Robot 2.

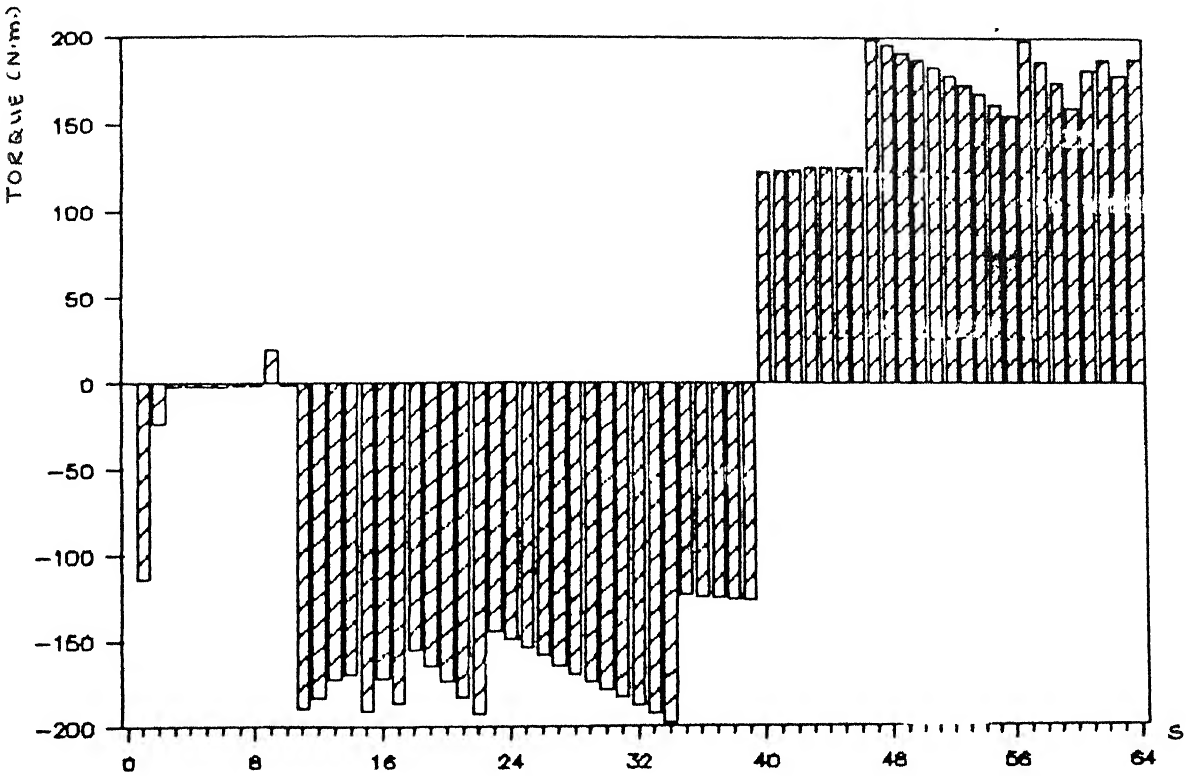


Fig. 4.4 : Joint 1-Nominal torque plot of Robot 1. with obstacle.

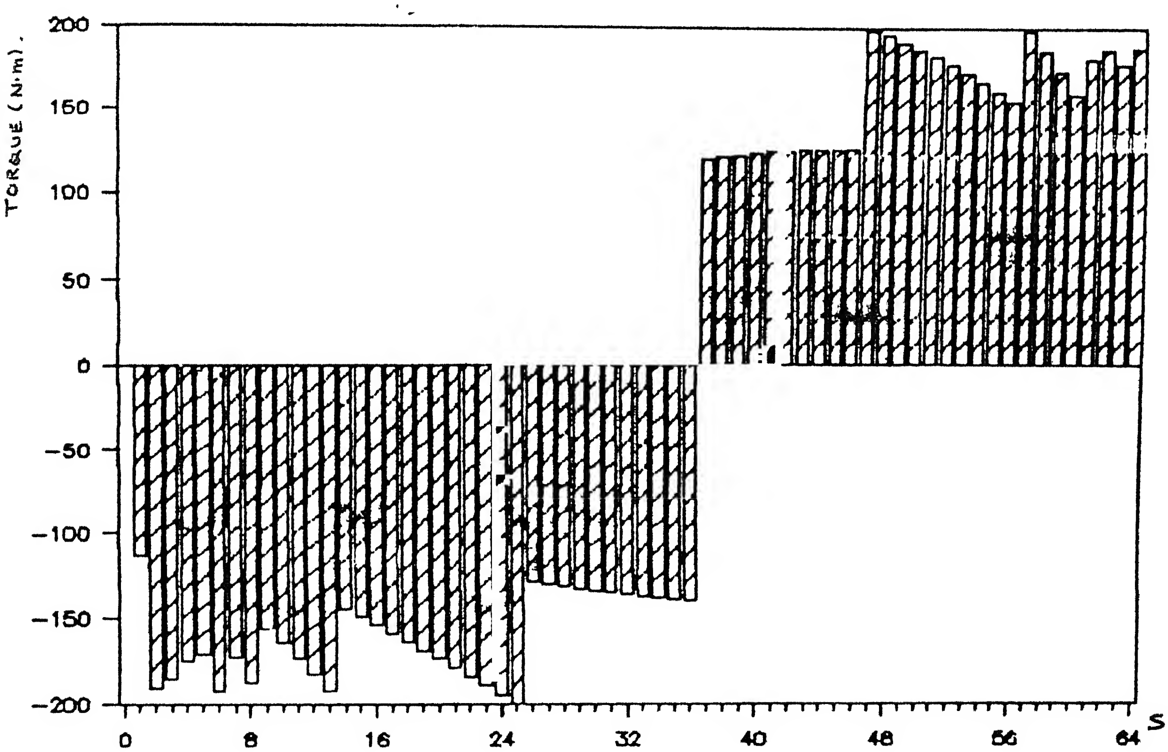


Fig. 4.5 : Joint 1-Optimal torque plot of Robot 1.

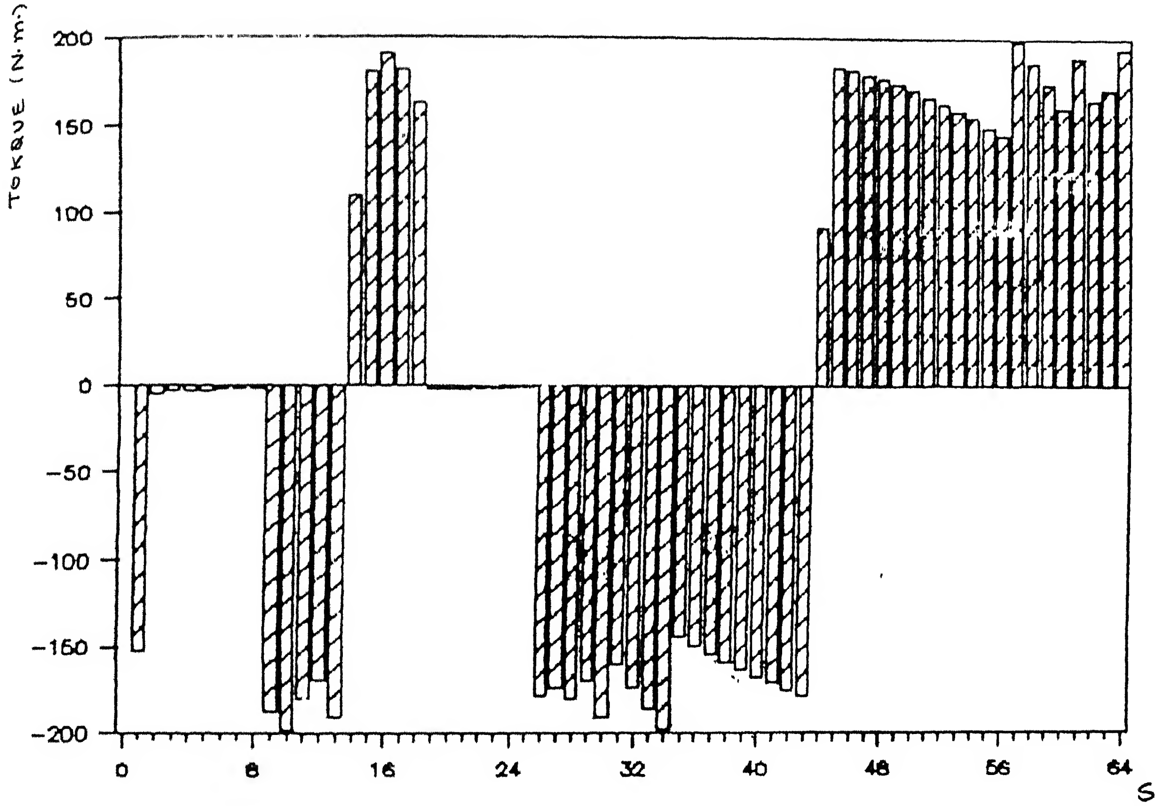


Fig. 4.6 : Joint 1-Optimal torque plot of Robot 2. with obstacle.

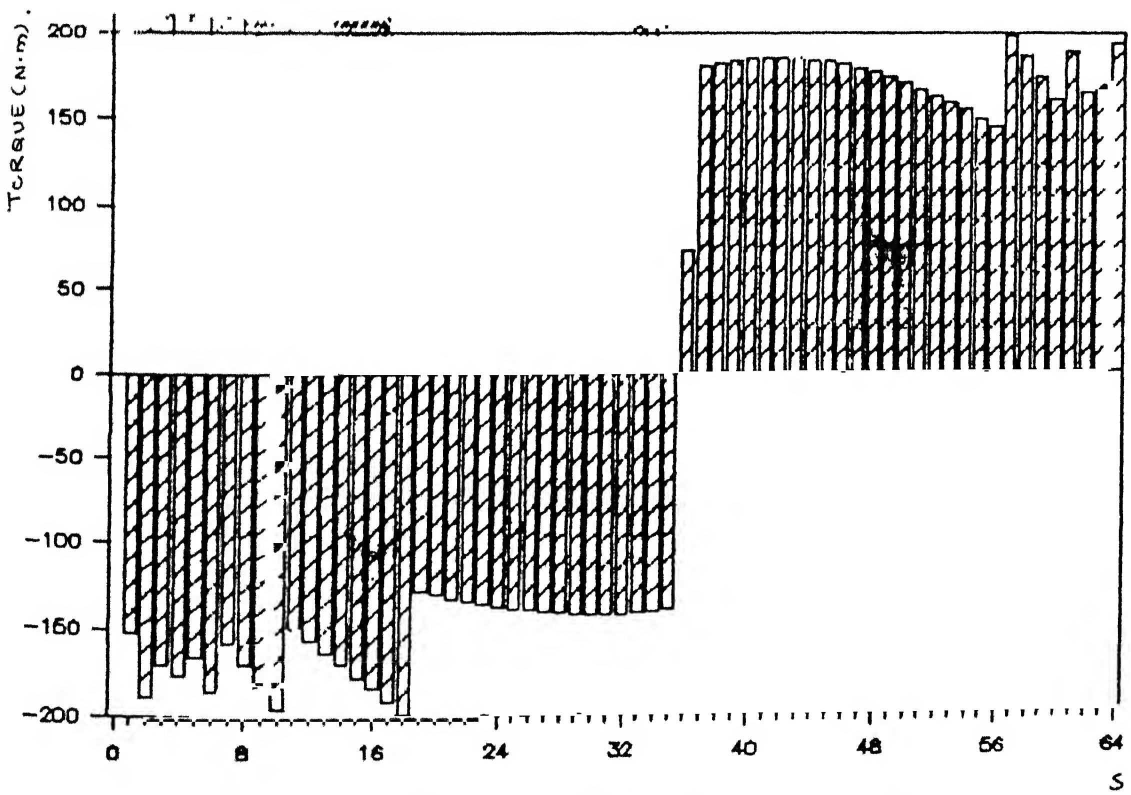


Fig. 4.7 : Joint 1-Optimal torque plot of Robot 2.

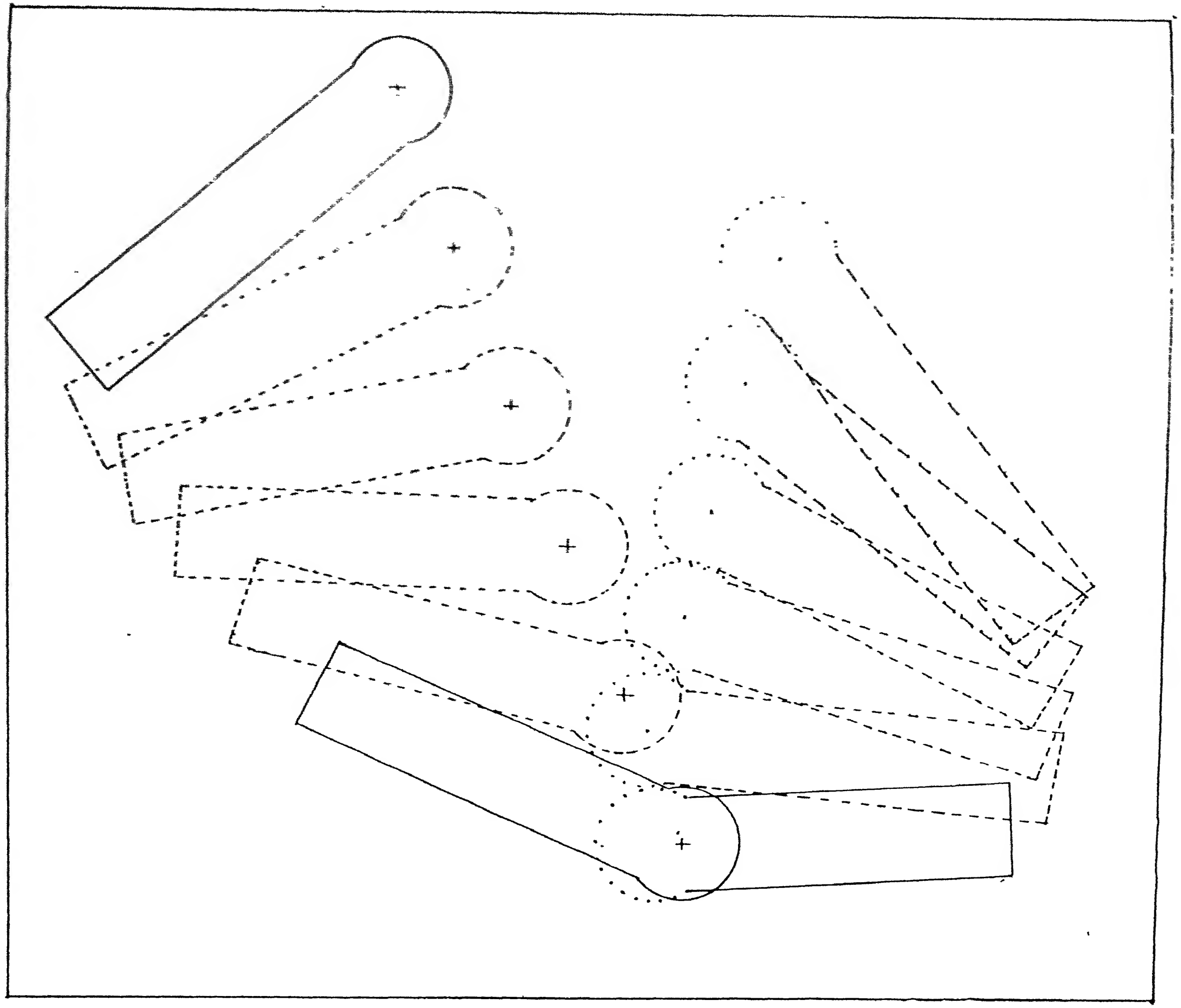


Fig. 4.8 : X-Y plot of the nominal trajectories of the two arms.

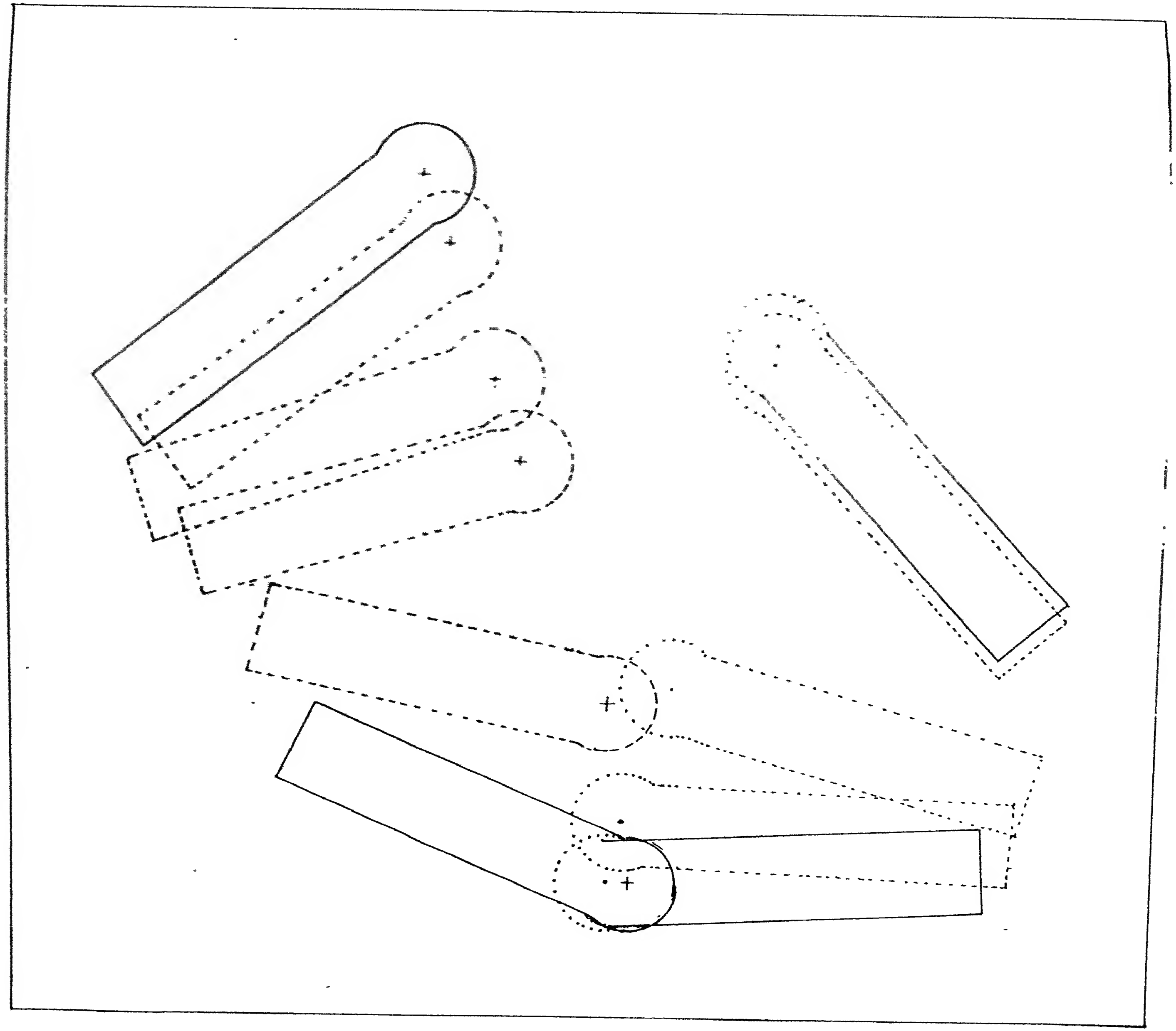


Fig. 4.9 : X-Y plot of the optimal trajectories of the two arms.

This fact is further reinforced by Figs. 4.4 and 4.7. The efficacy of the collision avoidance scheme is clearly brought out through Figs. 4.8 and 4.9 respectively.

CHAPTER 5

NEAR MINIMUM-TIME-ENERGY GEOMETRIC PATH FOR A SINGLE ROBOT

In the present industrial environment where robotic manipulators are finding widespread use, optimum utilization of robots attains significant dimensions. Some of the important considerations are minimizing energy or maximizing speed of operation. Maximizing speed of operation results in lower operation time and greater throughput. Minimizing energy results in reducing wear and tear of actuators as well as operational cost. But invariably, increasing the speed of operation entails greater consumption of energy. Thus, the requirement of minimum operating time and minimum energy are conflicting in nature. In general, the actuators are required to be used to their fullest extent while simultaneously, the wear and tear as well as energy consumption must be low.

The importance of time-optimal control of robotic manipulators has generated a wide research in this area. But most of the methods published so far fall short of requirement when the objective is neither minimum-time nor minimum energy, but a combination of both. Most of the methods deal with minimizing time given a particular path, but when the problem is to a path along which the cost-comprising of time and energy is minimum, then these methods are inadequate.

5.1 Problem Formulation :

In the MLV algorithm for finding minimum-energy paths, time was treated as a fixed parameter and the joint angles were treated as stage variables(Chapter 2). In contrast, time was a free variable in the problem of finding minimum time along a given path(Chapter 4). For the problem of finding near-minimum-time-energy geometric path, the objective function is a combination of total energy of the system and the time of traversal from system initial point to the final point. To accommodate this type of objective functions, the algorithm has to be suitably modified.

Given the initial configuration and final configuration of the robotic arm, a feasible trajectory and a nominal time of traversal is to be chosen first. The initial trajectory is divided into N subintervals and the subinterval costs computed and stored. In the first pass of the algorithm, variations in joint angles(state variables) are performed and a new trajectory is chosen such that the overall energy is lowered. This is similar to the method adopted in chapter 2. During this process, time is treated as a fixed parameter as variations in joint position do not affect the time but affects only the velocities and accelerations of individual joints. After each state variable is varied in all the subintervals, the resulting trajectory is parameterised and MLV applied to it as in Chapter 4. Each variation in parameter affects the time involved in that particular subinterval as well as the next

subinterval. This variation in time alters the Kinetic Energy for that interval and the next interval. Hence, before accepting a variation, it is necessary to check for reduced overall cost and not just reduction in time.

Specified the initial point (x_i, y_i, z_i) and final point (x_f, y_f, z_f) through which the manipulator has to pass, first a nominal feasible trajectory is chosen. This trajectory is then divided into N+1 equidistant points and the joint angles (q_1, q_2, q_3) at each of these points computed and stored. An initial traversal time is assumed, say 'T' secs. The time involved in each subinterval is taken to be T/N seconds.

In the first pass of MLV q_1, q_2, q_3 are treated as state variables. The velocities and accelerations are calculated at each of the N+1 points by means of difference equations

$$\dot{q}_i[k] = (q_i[k] - q_i[k-1])/Time[k] \tag{5.1}$$

$$\ddot{q}_i[k] = (\dot{q}_i[k] - \dot{q}_i[k-1])/Time[k] \tag{5.2}$$

for $i = 1, \ldots, 3$.

The MLV is then applied as in Chapter 2. and at each variation of q_i, checks are made to verify if this new q_i results in lowering of overall cost. If it does, and if the new q_i does not violate either the joint constraint or the torque constraint then q_i is replaced by this new value. At the end of this process, when q_1, q_2, q_3 have been varied in all the N subintervals, a new trajectory is available which is closer to the minimum-energy trajectory compared to the nominal trajectory. This trajectory is now parameterised.

This is achieved by expressing the joint angles as a function of a parameter 's'. The trajectory is traced out as s varies from 0 to s_{max}.

$$q_i = f_i(s) \qquad 0 \leq s \leq s_{max} \tag{5.3}$$

for i = 1,..,3.

Then $\dot{q}_i$ and $\ddot{q}_i$ can be expressed in terms of s, as

$$\dot{q}_i = \frac{df_i}{ds} \cdot \frac{ds}{dt} = \frac{df_i}{ds}\mu \tag{5.4}$$

$$\ddot{q}_i = \frac{df_i}{ds}\mu + \frac{d^2 f_i}{ds^2}\mu^2 \tag{5.5}$$

where $\mu = ds/dt$.

The dynamical equation for a 'n' joint manipulator in Lagrange-Euler form is given as

$$u_i = \sum_{j=1}^{n} D_{ij}\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} H_{ijk}\dot{q}_j\dot{q}_k + G_i \tag{5.6}$$

for i= 1,..n.

substituting eqns. (5.4) and (5.5) in eqn. (5.6)

$$u_i = M_i\mu + Q_i\mu^2 + G_i \; ; \; i = 1,..,n \tag{5.7}$$

$$\text{where } M_i = \sum_{j=1}^{n} D_{ij}\frac{df_j}{ds} \tag{5.8}$$

$$Q_i = \sum_{j=1}^{n}\sum_{k=1}^{n} H_{ijk}\frac{df_j}{ds}\frac{df_k}{ds} + \sum_{j=1}^{n} D_{ij}\frac{d^2 f_j}{ds^2} \tag{5.9}$$

Now the problem reduced to that of finding μ and μ for each interval such that the overall cost is reduced. Now, time for each subinterval is given by $2/(\mu[k]+\mu[k-1])$. Hence varying 'μ' affects the time for that

subinterval as well as the next, which in turn affect the KE. The problem , however, is to correlate changes in energy to variations in μ. This is done as follows.

Since M_i, Q_i, G_i are already calculated and stored the easiest way is to use these values to find a relationship between 'μ' and the energy. For simplicity let n be 3. From eqn. (5.8)

$$M_1 \frac{df_1}{ds} + M_2 \frac{df_2}{ds} + M_3 \frac{df_3}{ds} = D_{11}\left[\frac{df_1}{ds}\right]^2 + D_{22}\left[\frac{df_2}{ds}\right]^2 + D_{33}\left[\frac{df_3}{ds}\right]^2$$

$$+(D_{12}+D_{21}) \frac{df_1}{ds} \frac{df_2}{ds} + (D_{13}+D_{31}) \frac{df_1}{ds} \frac{df_3}{ds} + (D_{13}+D_{31}) \frac{df_2}{ds} \frac{df_3}{ds} \tag{5.10}$$

$$\text{i.e.,} \left[M_1 \frac{df_1}{ds} + M_2 \frac{df_2}{ds} + M_3 \frac{df_3}{ds}\right] \left[\frac{ds}{dt}\right]^2$$

$$= D_{11} (\dot{q}_1)^2 + D_{22} (\dot{q}_2)^2 + D_{33} (\dot{q}_3)^2 + (D_{12}+D_{21}) \dot{q}_1\dot{q}_2 + (D_{13}+D_{31}) \dot{q}_1\dot{q}_3 + (D_{23}+D_{32}) q_2q_3$$

$$= \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \dot{q}_3 \end{bmatrix} \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$= V^T D V \quad = \text{K.E.} \tag{5.11}$$

Since L.H.S. of (5.11) is precalculated, the variation in 'μ' can be easily calculated without taking recourse to calculation of velocity, D,H,G matrices etc,. The overall problem formulation is as shown below.

5.1.1 Near-Minimum-Time-Energy Geometric Path Planning Problem:

The problem can now be stated as follows.

$$\text{Given} \quad q_i = q_i \ (\text{initial}) \quad \text{at } t = 0$$
$$= q_i \ (\text{final}) \quad \text{at } t = \tau \qquad (5.12)$$

for $i=1,..,n$ and the dynamic eqns. given by (5.6) or (5.7), find the optimal inputs u_i^*, traversal time T^* and the time history of the variables q_1^*, q_2^*, q_3^* such that the cost function given by (5.13) is minimized subject to the bounds expressed in eqn. (5.12).

$$C = K_1 \int_0^T dt + K_2 \int_0^T (\beta_1 V^T D V + \beta_2 PE) \, dt \qquad (5.13)$$

where K_1, K_2, β_1, β_2 are constants, V is the joint velocity vector and PE is the potential energy of the arm.

5.3 Numerical examples:

Consider the case when the robot manipulator tip is required tomove from a point (0.5,0.2,1.2) to a point (0.6,-0.2,1.0) in the Cartesian coordinate space in near minimum - time - energy path. For simplicity, the path is assumed to be obstacle free. The manipulator arm is required to operate within the bounds on the input torques. These bounds are as chosen in example 2.1 .

Since the problem is to find the optimum energy and time of traversal, the cost function selected is of the form given in eqn. 5.13 . K_1 and K_2 are taken as 100 and 0.25 and β_1 and β_2 are taken as 15 and 0.5 respectively.

Equal weightage is assumed for the energy of the robot, as well as, for the time of traversal. Thus the choice of K_1 and K_2 is made such that both the terms being integrated in eqn. 5.13 are approximately of the same magnitude. This ensures equal priority to reduction in time, as well as, energy of the robot's path.

In order to study the dependence of the optimal path on the starting trajectory, various starting trajectories have been tried out. They are as follows :

case(i): Circular trajectory: The initial path chosen is such that its projection on X-Y plane is a semi-circle and movement in the vertical plane is in equal decrements.

case(ii): Starting trajectory close to minimum energy path: The nominal trajectory chosen in case(i) is minimized by application of the MLV algorithm. The criterion function is same as that chosen in example 2.1 . The algorithm was terminated when the resulting trajectory was sufficiently close to the optimum. The resulting trajectory was minimized according to problem formulated in 5.1 .

case(iii) The minimum-energy path is first obtained and then the procedure outlined in 5.1 applied to obtain the near - minimum - time - energy geometric path.

case(iv) The nominal path was assumed to a straight line joining the initial and the final points of the robot's desired motion.

TABLE 5.1

case	TIME		ENERGY		TOTAL COST	
	initial	optimal	initial	optimal	initial	optimal
i	1.0	0.4662	149.432	221.173	137.358	101.909
ii	1.0	0.2841	-	119.547	-	58.297
iii	1.0	0.2314	78.9399	88.6534	119.735	45.303
iv	1.0	0.2320	79.1770	99.9889	119.794	48.197

In all the above cases, the nominal time for the robot to move from the initial to the final configuration was assumed to be 1 second. The results obtained are presented in table 5.1 . The nominal trajectories of the first three joints and the optimal trajectories obtained fom case(iii) are shown in Figs. 5.1 to 5.3 . The optimal μ - s plot is shown in Fig. 5.4 . The torque plots of the first three joints are shown in Figs. 5.5 to 5.7 .

From the figures it is evident that while the joint trajectories for the near - minimum - time - energy geometric path are lines of minimum curvature close to the corresponding minimum - energy path, the torque plots show that the joint torques are bang bang in nature to ensure high velocities for individual joints. This results in a high speed of traversal and thus lower traversal time. Further, it can be seen that the final energy of the robot is higher than the energy along the nominal path. This is because the time of traversal has been reduced. Initially, the traversal time was 1 second but the optimal traversal time is 0.23 seconds. This reduction in time causes increase in the velocity of the arm which leads to increase in kinetic energy.

5.4 Conclusions :

In this chapter the near - minimum - time - energy problem was formulated. The problem formulation was tested with different nominal trajectories. From the results it is clear that the optimal trajectory will be in the

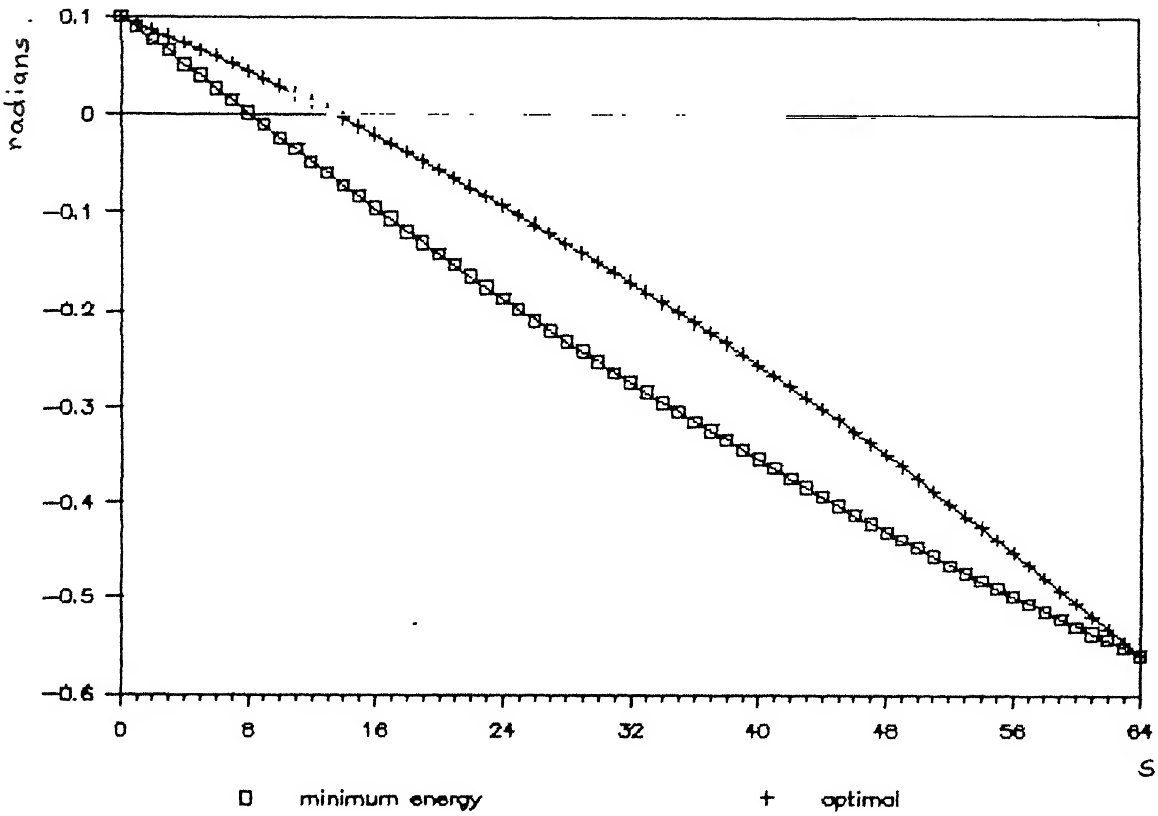


Fig. 5.1 : Joint 1 position plot for case (iii).

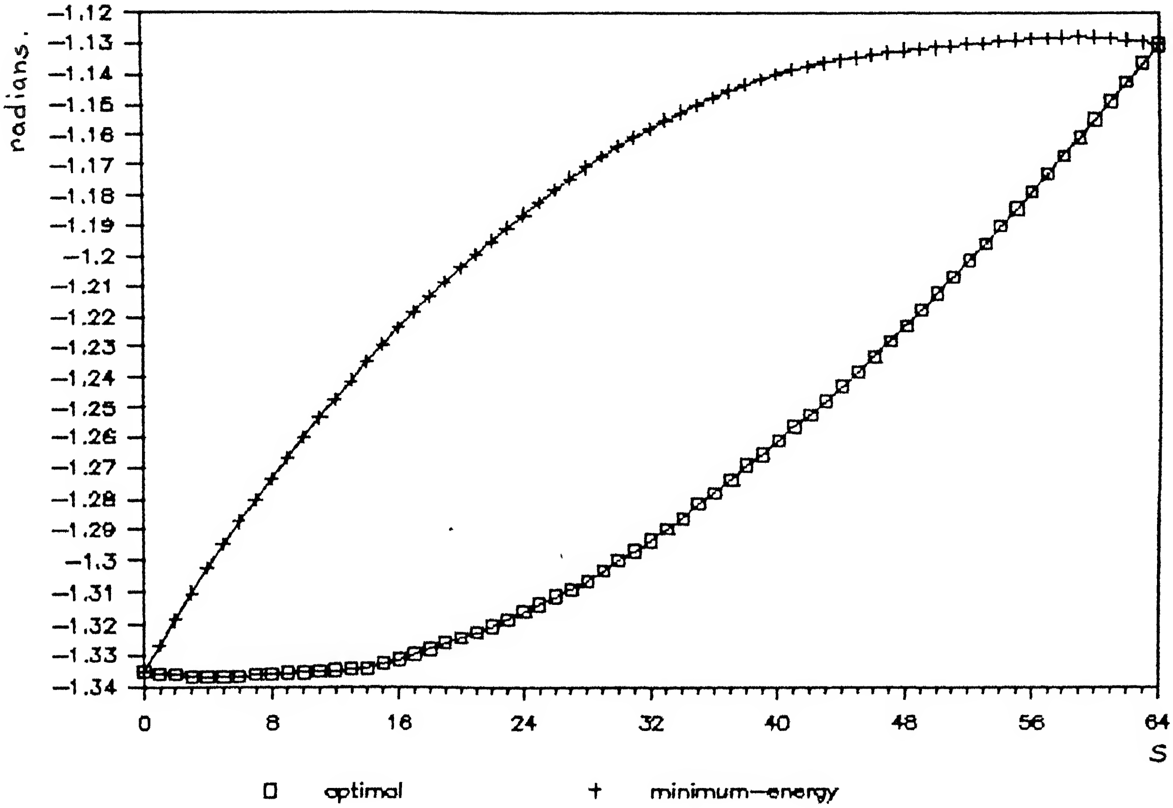


Fig. 5.2 : Joint 2 position plot for case (iii).

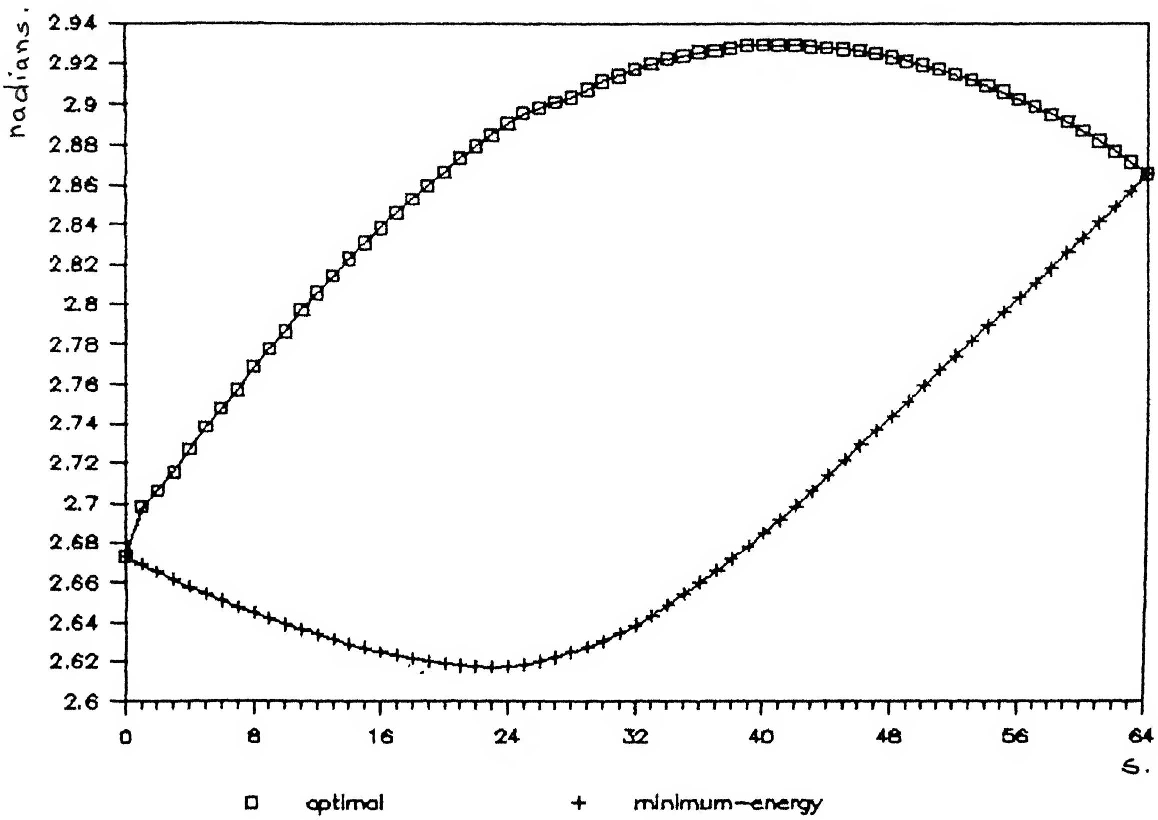


Fig. 5.3 : Joint 3 position plot for case (iii).

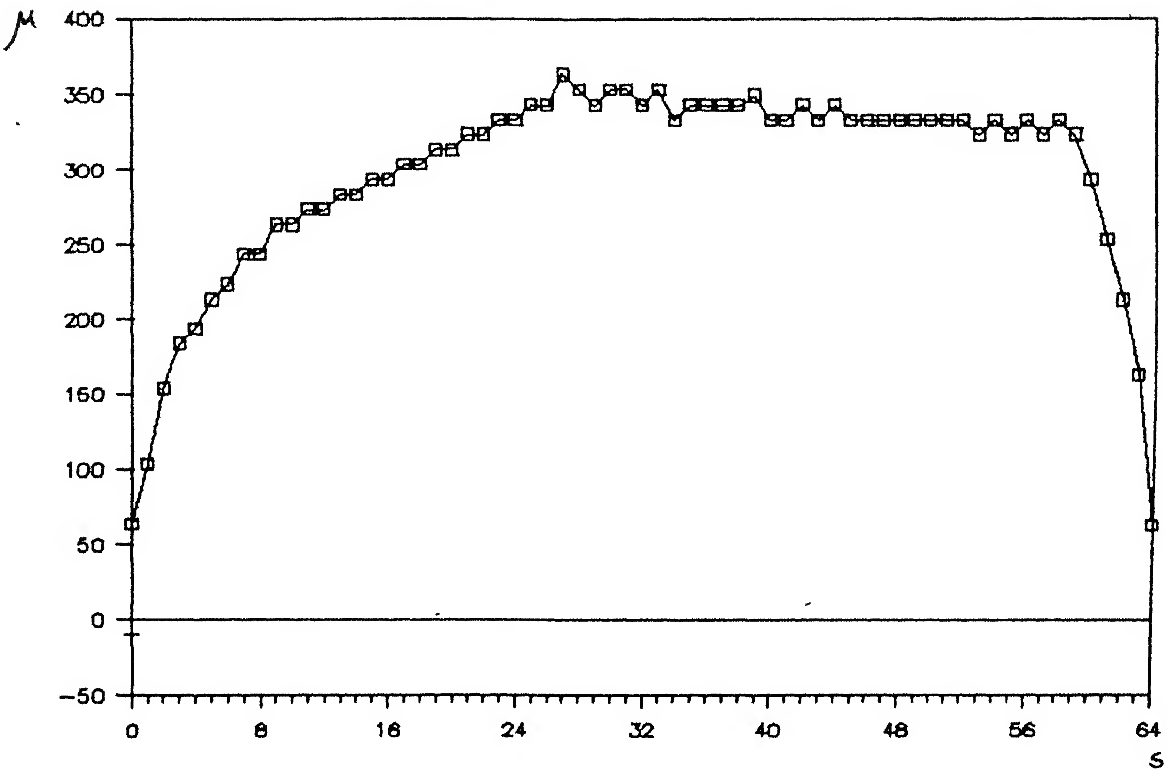


Fig. 5.4 : Phase-plane plot for case (iii).

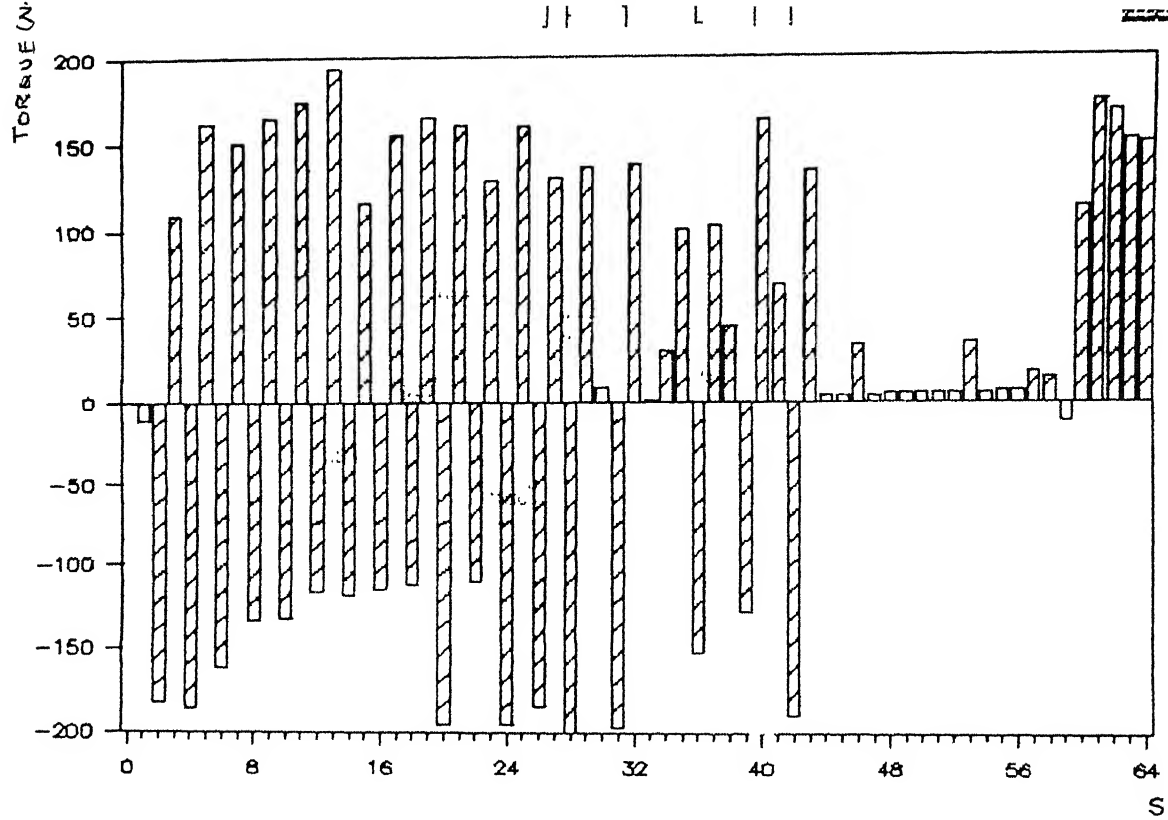


Fig. 5.5 : Optimal Joint 1 torque for case (iii).

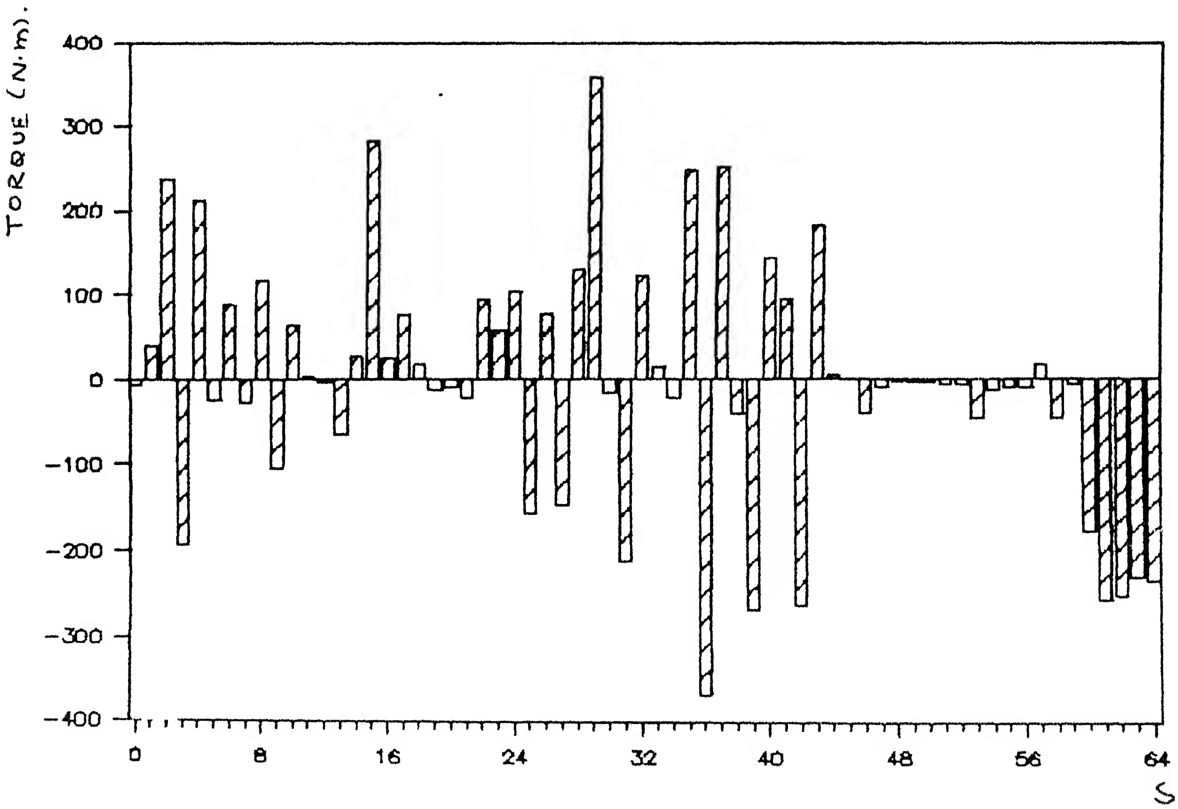


Fig. 5.6 : Optimal Joint 2 torque for case (iii).

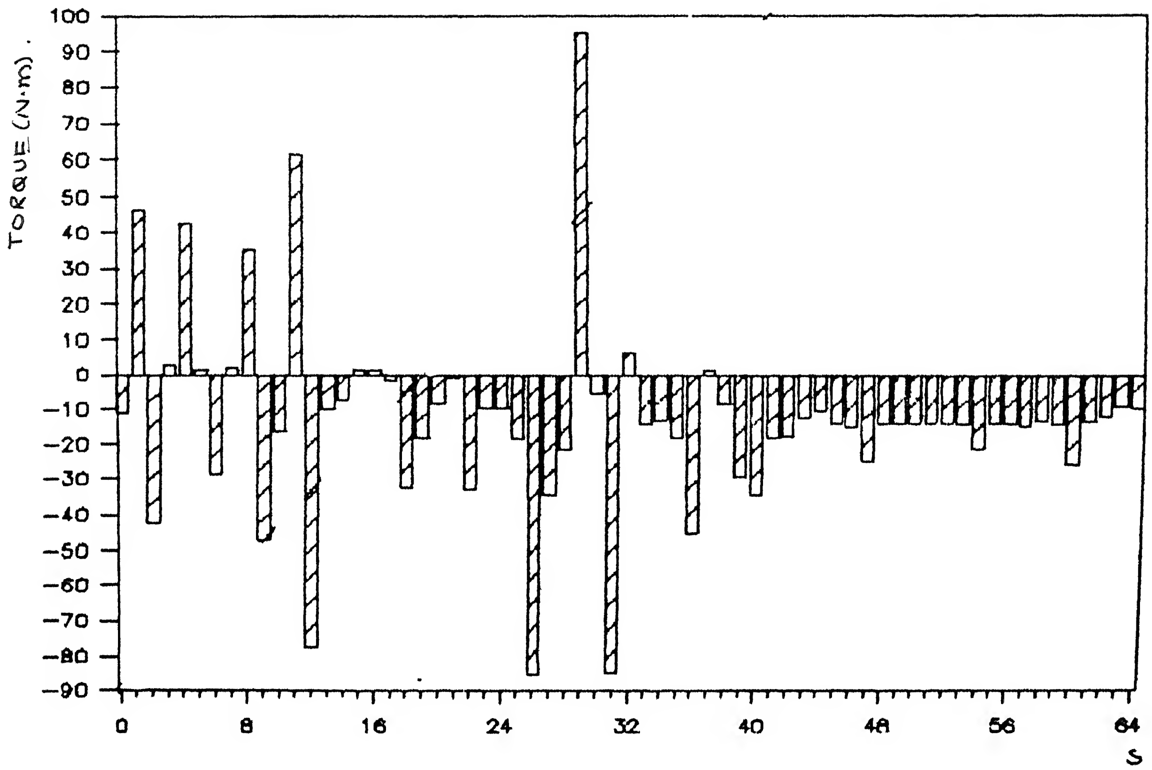


Fig. 5.7 : Optimal Joint 3 torque for case (iii).

neighbourhood of the minimum energy path. Further, if the starting trajectory is far off from the optimum, then the resulting trajectory may converge to a local optima. Thus, in order to obtain a near - minimum - time - energy geometric path it is desirable to choose the minimum - energy trajectory as the nominal trajectory.

CHAPTER 6.

GENERAL CONCLUSIONS AND SCOPE FOR FURTHER WORK.

In this thesis, methods for detecting collisions between different objects has been developed. This has been incorporated in the MLV algorithm and different types of robot path planning problems solved. This chapter lists out the conclusions drawn from the simulation studies. Scope for further work is also discussed.

Conclusions:

(1) The efficacy of collision detection schemes for the static, as well as the dynamic, obstacles has been clearly demonstrated by means of examples. The geometric checking scheme is powerful and can handle obstacles of different types.

(2) Minimum - Time and Minimum - Energy path planning for two robots has been successfully attempted for the first time.

(3) Method to formulate and solve minimum - time - energy path planning problems has been presented and guidelines for choosing such paths laid down.

SCOPE FOR FURTHER WORK :

Although simplification in the robot arm structure is made for the purpose of collision detection, the computation necessary is still large. This can be overcome by checking for

collision only at a discrete number of points on the trajectory. The minimum distance between the arm and the obstacle can be stored at each variation. At the next variation, checks are made only if this distance is lesser than a predetermined value. If the incremental step size is small, then the resulting variation in position of the arm will not be much and collision checks can be made only if the arms are very close toeach other. Another approach to the problem can be by mapping the arm and the obstacle positions in the joint space. If a computationally efficient method is devised, then the collsion checks can be easily performed by checking if the joint vector after any variation lies outside the feasible domain of the joint space.

When collision - free optimal paths for two robots are desired, then collision checking computation can be substantially reduced by determining a subset of the total workspace where collisions can occur. Only in this subset, detailed collision checks are performed. The algorithm can also be suitably modified to take into cosideration payload and parameter uncertainities.

REFERENCES.

(1) Shin.K.G and McKay.N.D, 'Selection of near-minimum time geometric paths for robotic manipulators ', IEEE Trans. Automat. Control, Vol. AC-31, pp. 501-511, June 1986.

(2) Lozano Perez.T , 'Automatic planning of manipulator transfer movements ', IEEE Trans. Sys. Man and Cyber., Vol. SMC 11,pp.687-698, Oct.1981.

(3) Lozano Perez.T , 'Spatial planning - A configuration space approach ', IEEE Trans. on Computers, Vol. C-32, pp.108-120,Feb.,1983.

(4) Lozano Perez.T , 'A simple motion planning algorithm', IEEE J. of Robotics and Automat., V-RA 3, pp. 224-237, June 1987.

(5) Lozano Perez.T and Wesley.M.A , 'An algorithm for planning collision free paths among polyhedral obstacles ',IEEE Tr. on Sys., Man and Cyber., SMC 17, pp. 21-31, Jan. / Feb. 1987.

(6) Brooks. R.A and Lozano Perez.T ,' A subdivisional algorithm in configuration space for find path with rotation ', IEEE Tr. on Sys., Man and Cyber., Vol. SMC 15, pp. 224-233, Mar. / Apr. 1985.

(7) Boyce. J.W ,'Interference detection among solids and surfaces ', Comm. of ACM, Vol. 22, No. 1, pp. 3-9, Jan. 1979.

(8) Brooks. R.A, ' Planning collision free motions for pick and place operations' , Int. J. of Robotics Res., Vol. 2, No. 4, pp.19-44, Fall 1983.

(9) Brooks. R.A, 'Solving find path problem by good representation of free space' , IEEE Tr. on Sys., Man and Cyber., Vol. SMC 13, pp 190-196, Mar. / Apr. 1983.

(10) Chien. R.T et.al., 'Planning collision free paths for robotic arm among obstacles' , IEEE Tr. on Patt. Anal. and Mach. Intell.,Vol. PAM 16, pp 90-96, Jan. 1984.

(11) Hasegawa and Terasaki, 'Collision avoidance ',IEEE Tr. on Sys., Man and Cyber., Vol. SMC 18, pp.337-347, May / June 1988.

(12) Lee. B.H and Lee. C.S.G, 'Collision free motion planning for two robots' , IEEE Tr. on Sys., Man and Cyber.,Vol. SMC 17, pp. 15-32, Jan. / Feb. 1987.

(13) Nagata. T et.al., 'Multi robot plan generation in a continuous domain - Planning by use of plangraph and avoiding collisions' ,IEEE J. of Robotics and Automat.,

Vol. 4, No. 1, pp.2-13, Feb. 1988.

(14) Gilbert E.G. and Johnson. D.W., 'Distance functions and their application to robot path planning in presence of obstacles, IEEE J. of Robotics and Automat., V - RA1, pp. 21-30 ,Mar. 1985.

(15) Gilbert E.G. and Johnson D.W., 'A fast procedure for computing the distance between complex objects in 3-D space' , IEEE J. of Robotics and Automat., Vol.4, No.2, pp.193-203 , Apr. 1988.

(16) Suh S.H. and Shin K.G., 'A variational dynamic programming approach to trajectory planning with distance safety criterion' , IEEE J. of Robotics and Automat., Vol. 4, No. 3, pp. 334-339, June 1988.

(17) Kahn M.E. and Roth B., 'Near minimum time control of open loop articulated kinematic chains' , ASME J. of Dynamic. Sys. Meas. and Control, pp. 164-172, Sept.1971

(18) Luh J.Y.S. and Walker W.M., 'Minimum time along the path for a mechanical arm' , Proc. 16th Conf. Decision and Control, pp.755-759, Dec.1977.

(19) Luh J.Y.S. and Lin C.S., 'Optimum path planning for mechanical manipulators' , ASME J. of Dynamic. Sys. Meas. and Control, pp 142-151, June 1981.

(20) Balamuraleedhar P., 'Minimum time trajectory planning for robot manipulators' , M.Tech thesis, Dept. of Electrical Engineering, IIT Kanpur, Jan. 1987.

(21) Hollerbach J.M., 'Dynamic scaling of manipulator trajectories' , ASME J. of Dynamic. Sys. Meas. and Control, pp. 102-106, Mar. 1984.

(22) Bobrow J.F., 'Optimal robot path planning using minimum time criterion', IEEE J. of Robotics and Automat., Vol. 4, No. 4, pp.443-450, Aug. 1988.

(23) Bobrow J.F. et. al., 'Time optimal control of robotic manipulator along specified paths' , Int. J. of Robotics Res., Vol. 4, No. 3, pp. 3-17, Fall 1985.

(24) Shin K.G. and McKay N.D., 'Minimum time control of robotic manipulator with geometric path constraints' , IEEE Tr. on Automat. control, Vol. AC 30, No. 6, pp. 531-541, Jun. 1985

(25) Shin K.G. and McKay N.D., 'A dynamic programming approach to trajectory planning of robotic manipulators' , IEEE Tr. on Automat. Control, Vol. AC 31, No. 6, pp. 191-200, Jun. 1986.

(26) Patrikar A.M., 'Optimal path planning of robot manipulators by the method of local variations ', M.Tech. Thesis, Dept. of Electrical Engineering, IIT Kanpur, Dec. 1987.

(27) Chernous'ko F.L., 'A local variation method for numerical solution of variational problems' , U.S.S.R Computational Mathematics and Mathematical Physics' , Vol. 5, No. 4, pp.234-242, 1965.

(28) Krylov L.A. and Chernous'ko F.L., 'Solution of problems of optimal control by the method of local variations' , U.S.S.R Computational Mathematics and Mathematical Physics, Vol. 6, No. 2, pp. 12-31, 1966.